L Number	Hits	Search Text	DB	Time stamp
1	0	(generat$3 with abort with code) and (error with recovery with disabled)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 14:36
2	0	(generat$3 with abort with code) and (recovery with disabled)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 14:38
3	43	generat$3 with abort with code	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 14:42
4	80	generat$3 with abort$3 with code	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 15:23
5	2	(generat$3 with abort$3 with code) and (recovery with enabl$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 14:43
11	80	error with recovery with enabled	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 15:13
12	24	(error with recovery with enabled) and (abort$3 with code)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 15:22
13	0	(generat$3 with abort with code) and (recovery with disabl$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 15:23
14	4	(generat$3 with abort$3 with code) and (717/$.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:02
15	7	creat$3 with abort$3 with code	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:06
16	514	error with recovery with enabl$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:12
17	2	(error with recovery with enabl$3) and (717/$.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:11
18	116	error with recovery with disabl$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:12
22	0	(error with recovery with disabl$3) and (717/$.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:12

23	0	(error with recovery with disabl$3) and (717/$.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:13
24	31	(error with recovery with disabl$3) and (714/$.ccls.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:19
25	2	run$4 with program with recovery with disabl$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:19
26	231	717/124.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:56
27	145	717/130.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:56
28	179	717/131.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:56
29	36	717/132.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:56
30	551	714/48.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 16:59
31	149	714/52.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 17:00

-	100	("5278840" "5793943" "6185696" "5634008" "6167532" "5581696" "6085029" "5214652" "5566298" "4785417" "6058268" "6505296" "5640503" "5925125" "5835761" "5835695" "6026016" "4989145" "5333308" "5689636" "5937331" "6161196" "4974248" "5204968" "5434805" "5444656" "5390323" "4821267" "5276692" "6405132" "5490250" "5321698" "5894583" "5997167" "3688274" "3795916" "4559596" "4852092" "4974147" "4979105" "5008807" "5371884" "5379414" "5651124" "5694617" "5703877" "5768620" "5802258" "4545016" "4949238").pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 10:22
-	22	(software with recovery) and (error near5 recovery near5 enab$$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 10:40
-	0	(pars$3 with source with error) and (detect$3 near5 error near5 program$1) and (insert$3 with code with recovery)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 11:02
-	0	(pars$3 with source with error) and (detect$3 with error with program$1) and (insert$3 with code with recovery)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 11:02
-	0	(pars$3 with source with error) and (detect$3 with error with condition$1) and (insert$3 with code with recovery)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 11:02

-	514	enab$4 with error with recovery	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 12:32
-	24	5581696.URPN.	USPAT	2003/06/12 12:30
-	2	5987249.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 14:00
-	564	recover$3 with (error near5 condition)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 14:07
-	3	(recover$3 with (error near5 condition)) and ((detect$3 or identif$7) with (error near5 condition near5 test))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 14:02
-	3	recover$3 with (error near5 condition near5 test)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 14:09
-	98	generat$3 with error with recovery with code	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 14:09
-	0	(generat$3 with error with recovery with code) and (generat$3 with program with abort with code)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 14:11
-	0	(generat$3 with error with recovery with code) and (generat$3 with abort with code)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 14:13
-	1	(generat$3 with error with recovery with code) and (insert$3 near5 program$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 14:49
-	2	5966541.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 15:46
-	2	5511164.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 16:06
-	0	software with recovery same (fail$3 with program$4 with assertion$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 16:07
-	8	fail$3 with program$4 with assertion$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/12 16:07
-	2365426	enabl$3 or disabl$3 with (program$4 near5 assertions)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 09:04

-	0	(enabl$3 or disabl$3 with (program$4 near5 assertions)) and (recover$3 with (fail$3 near5 program$4 near5 assertion$1) with run$1time)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 08:44
-	0	(enabl$3 or disabl$3 with (program$4 near5 assertions)) and ((fail$3 near5 program$4 near5 assertion$1) with run$1time)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 08:44
-	0	(enabl$3 or disabl$3 with (program$4 near5 assertions)) and ((fail$3 with program$4 with assertion$1) with run$1time) and (error with recover$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 08:45
-	0	(enabl$3 or disabl$3 with (program$4 near5 assertions)) and ((program$4 with assertion$1) with run$1time) and (error with recover$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 08:45
-	8524	(enabl$3 or disabl$3 with (program$4 near5 assertions)) and (error with recover$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 08:45
-	325	(enabl$3 or disabl$3 with (program$4 near5 assertions)) and (software with error with recover$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 08:45
-	0	((enabl$3 or disabl$3 with (program$4 near5 assertions)) and (software with error with recover$3)) and ((source near5 program) with (program$4 near5 assertion$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 08:47
-	0	((enabl$3 or disabl$3 with (program$4 near5 assertions)) and (software with error with recover$3)) and ((source with program) with (program$4 with assertion$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 08:47
-	3	((enabl$3 or disabl$3 with (program$4 near5 assertions)) and (software with error with recover$3)) and (generat$3 with error with recovery with code)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 08:51
-	1	(enabl$3 or disabl$3) with (program$4 near5 assertions)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 09:12
-	2	5966541.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 10:24
-	9	("4951195" | "5241678" | "5428786" | "5488714" | "5546586" | "5615369" | "5675803" | "5732275" | "5764992").PN.	USPAT	2003/06/13 09:33
-	5	5966541.URPN.	USPAT	2003/06/13 10:02
-	3	"6176209"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 10:28
-	0	"921059461"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 10:26

-	2	"61288523"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 10:26
-	2	5511164.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 10:31
-	2	"10340207"	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/13 10:31

File 347:JAPIO Oct 1976-20[redacted] Feb(Updated 030603)
(c) 2003 JPO & JAPIO
File 350:Derwent WPIX 1963-2003/UD,UM &UP=200337
(c) 2003 Thomson Derwent

Set	Items	Description
S1	864	ASSERTION? ? OR ASSERT
S2	29472	(ERROR? ? OR FAIL? OR FAULT? ? OR PROBLEM? ? OR TROUBLE OR FLAW OR IRREGULAR? OR GLITCH?? OR FALSE OR ODD???? OR STRANGE-???? OR ANOMAL? OR ABNORMAL?)(5N)(TEST??? OR CHECK???)
S3	53140	(ERROR? ? OR FAIL? OR FAULT? ? OR PROBLEM? ? OR TROUBLE OR FLAW OR IRREGULAR? OR GLITCH?? OR FALSE OR ODD???? OR STRANGE-???? OR ANOMAL? OR ABNORMAL?)(5N)(CONDITION? ? OR STATE OR STATES OR SITUATION OR STATUS)
S4	518323	(RECOVER? OR CORRECT? OR FIX??? OR MEND??? OR REMED??? OR -RECTIF? OR REPAIR? OR PATCH? OR RESTOR? OR RESOLV? OR SOLV?)(-5N)(ENABL? OR ON OR DISABL? OR OFF)
S5	23251	(INSERT? OR PUT???? OR PLAC??? OR PLACEMENT OR ADD??? OR APPEND?)(5N)(CODE? ? OR INSTRUCTION? ? OR FUNCTION? ? OR COMMAND? ? OR ROUTINE? ?)(5N)(PROGRAM? ? OR CODE OR APPLICATION? ? OR SOFTWARE)
S6	52605	(ERROR? ? OR FAIL? OR FAULT? ? OR PROBLEM? ? OR TROUBLE OR FLAW OR IRREGULAR? OR GLITCH?? OR FALSE OR ODD???? OR STRANGE-???? OR ANOMAL? OR ABNORMAL?)(10N)(PROGRAM? ? OR CODE OR APPLICATION? ? OR SOFTWARE OR INSTRUCTIONS OR OPERATIONS)
S7	2	S1 AND S4:S5 AND S6 AND IC=G06F
S8	10	S1 AND S4:S5 AND IC=G06F
S9	304	S2 AND S4:S5 AND S6 AND IC=G06F
S10	147	S2 AND S4 AND S6 AND IC=G06F
S11	18827	(WHEN OR IF OR SHOULD OR WHILE OR THAT)(5W)S4
S12	10	S10 AND S11
S13	31	S2 AND S3:S4 AND S5 AND S6 AND IC=G06F
S14	29	S13 NOT (S8 OR S12)
S15	157	S3 AND S4:S5 AND S6 AND IC=G06F
S16	110	S3 AND S4 AND S6 AND IC=G06F
S17	9	S3 AND S11 AND S6 AND IC=G06F
S18	4	S2 AND S3 AND S4 AND S5

8/5/1 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO

07129472 **Image available**
SYSTEM AND METHOD FOR PUBLICATION

PUB. NO.: 2001-357142 [JP 2001357142 A]
PUBLISHED: December 26, 2001 (20011226)
INVENTOR(s): TSUCHIDA SHINGO
APPLICANT(s): ECOS CORP
APPL. NO.: 2000-177001 [JP 2000177001]
FILED: June 13, 2000 (20000613)
INTL CLASS: **G06F-017/60**

ABSTRACT

PROBLEM TO BE **SOLVED** : To obtain a publication which **enables** a consumer to enjoy reading and has such interactivity that the consumer sends information by oneself.

SOLUTION: This system is equipped with a providing means 4 for a place of speech where a theme of debate or discussion is made open to the public on a communication network 2 and opinions, **assertions** , personal experienice, etc., are invited with the notice that a certain number or more of opinions, **assertions** , personal experience, etc., to the theme when contributed are published as a book without any correction, a contributed data storage means 5 which stores contributed data sent from an information processing terminal 3 connected through the communication network, and a publishing means 6 which creates a publication 7 by gathering the contributed data.

8/5/2 (Item 2 from file: 347)
DIALOG(R)File 347:JAPIO

04505826 **Image available**
GUARANTEEING CIRCUIT FOR RECOVERY TIME OF DMA HOLD ACKNOWLEDGEMENT SIGNAL

PUB. NO.: 06-149726 [JP 6149726 A]
PUBLISHED: May 31, 1994 (19940531)
INVENTOR(s): KIDO RYOJI
APPLICANT(s): MATSUSHITA ELECTRIC IND CO LTD [000582] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 04-297203 [JP 92297203]
FILED: November 06, 1992 (19921106)
INTL CLASS: [5] **G06F-013/28**
JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- Memory Units)
JOURNAL: Section: P, Section No. 1793, Vol. 18, No. 460, Pg. 130, August 26, 1994 (19940826)

ABSTRACT

PURPOSE: To guarantee the recovery time of the hold acknowledge signal that a bus arbitration control circuit outputs to a DMA controller when the DMA controller outputs a DMA bus cycle request signal.

CONSTITUTION: A signal DMARVW(3) is asserted at the end point of time of a CPU cycle, continued for a period satisfying the recovery time of the DMA hold acknowledgement signal, and then negated. When the DMA controller makes a request to acquire a bus during the **assertion** period of the DMARVW signal, the DMA hold acknowledgement signal corresponding to the bus request is pended (reception wait state). After the DMARVW signal is negated, the output of the DMA hold acknowledgement signal is **enabled** . The **recovery** of the DMA hold acknowledgement signal can be guaranteed by optional length corresponding to the purpose of use of a circuit designer by controlling the **assertion** period of the DMARVW signal.

8/5/3 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX

014795722 **Image available**
WPI Acc No: 2002-616428/200266
Related WPI Acc No: 2003-091515
XRPX Acc No: N02-487699

Fault tolerant operation method for processing system, involves halting processor operation when error is irrecoverable and restoring master and shadow processors to saved state when error is recoverable

Patent Assignee: COMPAQ COMPUTER CORP (COPQ)
Inventor: BRUCKERT W F; JARDINE R L; KLECKA J S
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6393582	B1	20020521	US 98209635	A	19981210	200266	B

Priority Applications (No Type Date): US 98209635 A 19981210
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6393582	B1		10	G06F-011/34	

Abstract (Basic): US 6393582 B1

NOVELTY - Master processor address and data signals are checked against shadow processor address and data signals to **assert** a diverge signal during mismatch. Master processor (12) detects generation of error in master or shadow processor (14) and halts processor operation when the error is irrecoverable. Processor state and data are stored in the memory and the processors are restored to saved state when the error is recoverable.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:

(1) Article of manufacture comprising storage medium storing fault tolerant operation program; and

(2) Computer system.

USE - For processing system.

ADVANTAGE - **Enables recovering** smoothly and quickly from self checking divergence of pairs of self checking processor modules during the generation of error.

DESCRIPTION OF DRAWING(S) - The figure shows the simplified block diagram of the logical processor.

Master processor (12)
Shadow processor (14)
pp; 10 DwgNo 1/5

Title Terms: FAULT; TOLERATE; OPERATE; METHOD; PROCESS; SYSTEM; HALT; PROCESSOR; OPERATE; ERROR; RESTORATION; MASTER; SHADOW; PROCESSOR; SAVE; STATE; ERROR; RECOVER
Derwent Class: T01
International Patent Class (Main): **G06F-011/34**
File Segment: EPI

8/5/4 (Item 2 from file: 350)
DIALOG(R)File 350:Derwent WPIX

014139885 **Image available**
WPI Acc No: 2001-624096/200172
XRPX Acc No: N01-464920

Counting rate adjusting method involves including design entity sequenced in accordance with design cycle in digital circuit design and incrementing counter in accordance with design cycle rather than sampling cycle

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: BARGH J F; HUNT B R; ROESNER W; WILLIAMS D E

Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6212491	B1	20010403	US 98190862	A	19981109	200172	B

Priority Applications (No Type Date): US 98190862 A 19981109
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6212491	B1		30	G06F-017/50	

Abstract (Basic): US 6212491 B1

NOVELTY - Design entity sequenced in accordance with the design cycle which is a multiple of the simulator cycle, is included in digital circuit design. Count event signal within the instrumentation entity is asserted in response to the occurrence of event within the design entity. Sampling signal having a cycle period equal to the period of design cycle is generated and compared with count event signal.

DETAILED DESCRIPTION - Sampling and count event signals are compared over the design cycle to determine the **assertion** of both the sampling and count event signals. Counter is incremented in accordance with the design cycle rather than the simulator cycle in response to the detection of **assertion** of both the sampling and count event signals. INDEPENDENT CLAIMS are also included for the following:

(a) Information handling system;
(b) Computer program;
(c) Counter

USE - For simulation of digital circuit design.

ADVANTAGE - Enables to monitor characteristics of specific modules or sub-modules of large scale design to efficiently and accurately diagonize problems and assess the **correctness** of overall design. **Enables** interactive design and simulation of complex circuits and systems, digital devices, modules and systems. Improves model build and simulation processes to allow the designer to easily instrument and monitor a simulation model. Utilizes instrumentation modules written in hardware description language to monitor the performance of computer generated digital circuit designs.

DESCRIPTION OF DRAWING(S) - The figure shows the flow diagram depicting model build process.

pp; 30 DwgNo 4D/6

Title Terms: COUNT; RATE; ADJUST; METHOD; DESIGN; ENTITY; SEQUENCE; ACCORD; DESIGN; CYCLE; DIGITAL; CIRCUIT; DESIGN; INCREMENT; COUNTER; ACCORD; DESIGN; CYCLE; SAMPLE; CYCLE
Derwent Class: T01; U11
International Patent Class (Main): **G06F-017/50**
International Patent Class (Additional): **G06F-009/455**
File Segment: EPI

8/5/5 (Item 3 from file: 350)
DIALOG(R)File 350:Derwent WPIX

014080389 **Image available**
WPI Acc No: 2001-564603/200163
Related WPI Acc No: 1999-580062
XRPX Acc No: N01-420296

Binary code test/protection/correction method for computer programs, rewrites binary code by installing binary software patches **into original binary code based** on **analysis of control flow representation**

Patent Assignee: AGARWAL A (AGAR-I); INCERT SOFTWARE CORP (INCE-N)
Inventor: AGARWAL A
Number of Countries: 001 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20010013119	A1	20010809	US 97985052	A	19971204	200163	B
			US 99358673	A	19990722		
US 6305010	B2	20011016	US 97985052	A	19971204	200164	

Priority Applications (No Type Date): US 97985052 A 19971204; US 99358673 A 19990722
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20010013119	A1		17	G06F-009/44	Cont of application US 97985052 Cont of patent US 5966541
US 6305010	B2			G06F-009/45	Cont of application US 97985052 Cont of patent US 5966541

Abstract (Basic): US 20010013119 A1

NOVELTY - A control flow representation of a binary code is generated and a binary software patches are defined. Installation area of software **patches** are determined based **on** the analysis of the control flow representation. The binary codes are rewritten into the original code by installing the binary software patches in determined installation area.

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for a data processing system.

USE - For type error problem detection, repair and testing, **assertion** checking, coverage testing, continuous testing, bootstrap regression testing, test path identification, zip code identification, telephone number or area code identification, currency identification, virus protection and corrupted argument protection.

ADVANTAGE - Since the binary codes are rewritten by installing binary software patches at selected locations determined by analysis of control flow representation, a new binary representation is produced in which the problem is fixed such that the code that operates on variables containing type error is converted into code that correctly accounts for the type error and allocates right sized data fields to store the resulting values and hence provides ultimate end-to-end test.

DESCRIPTION OF DRAWING(S) - The figure shows the program segment of binary code testing method.

pp; 17 DwgNo 1A/12

Title Terms: BINARY; CODE; TEST; PROTECT; CORRECT; METHOD; COMPUTER; PROGRAM; REWRITING; BINARY; CODE; INSTALLATION; BINARY; SOFTWARE; PATCH; ORIGINAL; BINARY; CODE; BASED; ANALYSE; CONTROL; FLOW; REPRESENT
Derwent Class: T01
International Patent Class (Main): **G06F-009/44 ; G06F-009/45**
File Segment: EPI

8/5/6 (Item 4 from file: 350)
DIALOG(R)File 350:Derwent WPIX

012773835 **Image available**
WPI Acc No: 1999-580062/199949
Related WPI Acc No: 2001-564603
XRPX Acc No: N99-428243

Binary code testing, correcting and protecting method for computer systems

Patent Assignee: INCERT SOFTWARE CORP (INCE-N)
Inventor: AGARWAL A
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5966541	A	19991012	US 97985052	A	19971204	199949	B

Priority Applications (No Type Date): US 97985052 A 19971204
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5966541	A		17	G06F-009/45	

Abstract (Basic): US 5966541 A

NOVELTY - A set of binary software patches (46) are defined and are

installed to rewrite original binary code on account of the installed software. The installation of software **patches** is based **on** the analysis of a control flow and data flow representation of a binary code.

DETAILED DESCRIPTION - The binary code is rewritten by installing binary software patches in original binary code. A value or variable is selected from the binary code and is tracked using the data flow representation. An INDEPENDENT CLAIM is also included for the processor system for testing, protecting or correcting binary code.

USE - For performing overflow detection, repair and test, **assertion** checking, coverage testing, continuous testing, bootstrap regression testing, argument remediation coverage testing, test path identification, date identification, zip code identification, telephone number or area code identification, currency identification, virus protection and corrupted argument protection.

ADVANTAGE - New remediated binary codes are produced by replacing old binary codes thereby date overflow problem is avoided and data is treated correctly. False occurrences of binary values are eliminated using multiple execution of interpretation program and thereby date fields are identified correctly.

DESCRIPTION OF DRAWING(S) - The figure is the flowchart for binary code correcting, testing or protecting procedure.

Binary software patches (46)

pp; 17 DwgNo 1B/12

Title Terms: BINARY; CODE; TEST; CORRECT; PROTECT; METHOD; COMPUTER; SYSTEM
Derwent Class: T01
International Patent Class (Main): **G06F-009/45**
File Segment: EPI

8/5/7 (Item 5 from file: 350)

012117334 **Image available**
WPI Acc No: 1998-534246/199846
XRPX Acc No: N98-416897

Handshaking circuit for establishing parent/child relationship between network nodes - has state detector monitoring transmission line; detects predetermined first and second states and contention when first states are simultaneously asserted

Patent Assignee: NEC CORP (NIDE); NIPPON ELECTRIC CO (NIDE)
Inventor: NYU T
Number of Countries: 026 Number of Patents: 004
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 872980	A2	19981021	EP 98105398	A	19980325	199846	B
JP 10271144	A	19981009	JP 9773187	A	19970326	199851	
KR 98080663	A	19981125	KR 9810334	A	19980325	200005	
JP 3161359	B2	20010425	JP 9773187	A	19970326	200126	

Priority Applications (No Type Date): JP 9773187 A 19970326
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 872980	A2	E	17	H04L-012/413	
Designated States (Regional): AL AT BE CH DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI					
JP 10271144	A		15	H04L-012/40	
KR 98080663	A			H04L-012/42	
JP 3161359	B2		15	H04L-012/40	Previous Publ. patent JP 10271144

Abstract (Basic): EP 872980 A

The circuit includes a state detector (3) monitoring the transmission line (2) that connects the nodes. This detects if the line is in a Parent-Notify or Child-Notify. A Root-Contention exists when both Notify states are simultaneously asserted on the line, and an idle state exists when either of the two states is not asserted.

A state machine (4) is coupled to the detector, and asserts one of

the Notify states on [redacted] line, initiating a handshaking [redacted]rocess, and relinquishes the Notify state when the contention is detected. A receive counter (10) and a transmit counter (11) counts times between contention and idle, and Notify and contention states. The receive and the transmit count values are compared, the state machine switches to ParentNotify if the time between the last detected Notify and last contention, is smaller than between the last contention and idle; it switches to ChildNotify if the reverse is true.

USE - For establishing parent-child relationship between adjacent nodes by **resolving** contention between them **on** , for example, serial bus, where they simultaneously **assert** same protocol status.

ADVANTAGE - Resolves internodal contention regardless of length of transmission medium.

Dwg.4/13

Title Terms: CIRCUIT; ESTABLISH; PARENT; CHILD; RELATED; NETWORK; NODE; STATE; DETECT; MONITOR; TRANSMISSION; LINE; DETECT; PREDETERMINED; FIRST; SECOND; STATE; CONTENTION; FIRST; STATE; SIMULTANEOUS

Derwent Class: W01

International Patent Class (Main): H04L-012/40; H04L-012/413; H04L-012/42

International Patent Class (Additional): **G06F-013/00** ; H04L-012/44

File Segment: EPI

8/5/8 (Item 6 from file: 350)

DIALOG(R)File 350:Derwent WPIX

009450685 **Image available**

WPI Acc No: 1993-144210/199317

XRPX Acc No: N93-110096

Data processing system memory resetting appts. - resets memory controller in same manner at initial system power- on **, at power** recovery **after power interrupt in which data has been preserved, and after power recovery in which data has been lost**

Patent Assignee: BULL HN INFORMATION SYSTEMS INC (HONE)

Inventor: BARLOW G J; BOWDEN R D; PENCE M A; SANFACON M E; SOMERS J S

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5204964	A	19930420	US 90593917	A	19901005	199317	B

Priority Applications (No Type Date): US 90593917 A 19901005

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5204964	A		18	G06F-012/16	

Abstract (Basic): US 5204964 A

The memory has memory elements, a refresh clock and a refresh counter for counting refresh cycles and providing refresh signals to the memory elements, the memory elements and refresh device are connected from the power system and from a battery back-up. A state detector is connected from the refresh counter for detecting a change in state of the refresh counter state equivalent to the reset state of the refresh counter and asserting a state change signal. A device responsive to the state change signal and to the occurrence of the reset signal provides a memory controller reset signal, so that the memory controller reset signal occurs in synchronisation with the change of state of the refresh counter to a state equivalent to the refresh counter reset state.

The memory rest further includes a time-out counter responsive to the **assertion** of the reset signal and to the refresh clock for counting refresh cycles in synchronisation with the refresh counter. A time-out detector is responsive to the time out counter for providing a time out signal when the time-out counter has counted a refresh period plus one clock cycle and to the state change signal for providing the memory controller reset signal when the time-out counter has counted a refresh cycle plus one clock period and the state change signal has not been asserted.

ADVANTAGE - Reliably avoids loss of memory data which has been preserved by battery back-up operation.

Dwg.3/3

Title Terms: DATA; PROCESS; SYSTEM; MEMORY; RESET; APPARATUS; RESET; MEMORY; CONTROL; MANNER; INITIAL; SYSTEM; POWER; POWER; RECOVER; AFTER; POWER; INTERRUPT; DATA; PRESERVE; AFTER; POWER; RECOVER; DATA; LOST

Derwent Class: T01; U24

International Patent Class (Main): **G06F-012/16**

File Segment: EPI

8/5/9 (Item 7 from file: 350)

DIALOG(R)File 350:Derwent WPIX

009081045 **Image available**

WPI Acc No: 1992-208467/199225

XRPX Acc No: N92-157976

Computer bus COM communication protocol - uses one type of device which only assert **their ID in one priority, and second type which practise round-robin arbitration**

Patent Assignee: CLEARPOINT RES CORP (CLEA-N)

Inventor: BAKER L D; HERBST W C; STEVENS G W

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5119292	A	19920602	US 89384824	A	19890721	199225	B

Priority Applications (No Type Date): US 89384824 A 19890721

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5119292	A			G06F-013/36	

Abstract (Basic): US 5119292 A

A first-type bus device is designed for use in a computer system having one or more second-type bus devices. The second-type devices practice a round robin arbitration scheme, but the first type devices do not. According to the round-robin scheme, a second-type device which wins an arbitration asserts its ID on the system bus, each second-type device seeking to arbitrate compares its ID with that asserted by the current bus master, and uses that comparison to determine whether to **assert** its ID during arbitration in a high or low priority manner.

The first-type bus device assures that first-type and second-type devices never arbitrate at the same time. During arbitration, first-type devices only **assert** their ID number in one priority. Unlike second-type devices, they cannot **assert** their ID either in a high or low priority manner. Each first-type device is given a selectable ID number, but when a first type device becomes bus master, it asserts an ID number which is independent of its selectable ID number. The ID number asserted could be the ID number of the first device to become bus master after the computer containing the first-type device is turned **on** . It also could be a **fixed** ID number which is fixed independently of the ID number selected for the first-type device.

USE - Computer bus communication protocols e.g. COM protocol which can arbitrate in either high or a low priority band and which perform round robin arbitration.

Dwg.24/28

Title Terms: COMPUTER; BUS; COMMUNICATE; PROTOCOL; ONE; TYPE; DEVICE; ID; ONE; PRIORITY; SECOND; TYPE; PRACTICE; ROUND; ROBIN; ARBITER

Derwent Class: T01

International Patent Class (Main): **G06F-013/36**

File Segment: EPI

8/5/10 (Item 8 from file: 350)

DIALOG(R)File 350:Derwent WPIX

007303744
WPI Acc No: 1987-300751/198743
XRPX Acc No: N87-224675

Computer system with source code re-creation capability - appends **compiled** code **information necessary to re-create source which generated compiled** code

Patent Assignee: TEXAS INSTR INC (TEXI)
Inventor: SRIVASTAVA A
Number of Countries: 004 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 243110	A	19871028	EP 87303392	A	19870416	198743 B
US 5249275	A	19930928	US 86854221	A	19860421	199340
			US 88191857	A	19880504	
			US 89316556	A	19890227	
			US 91696265	A	19910430	

Priority Applications (No Type Date): US 86854221 A 19860421; US 88191857 A 19880504; US 89316556 A 19890227; US 91696265 A 19910430
Cited Patents: 3.Jnl.ref; A3...9122; No-SR.Pub
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 243110	A	E	19		
Designated States (Regional): DE FR GB					
US 5249275	A		8	G06F-009/45	Cont of application US 86854221
					Cont of application US 88191857
					Cont of application US 89316556

Abstract (Basic): EP 243110 A

The method involves translating a source **code** statement into an object **code** block, **appending** to the block information sufficient to recreate the source **code** statement, and linking the object **code** block and **appended** information into a list with object **code** and **appended** information for any related source **code** statements. The above steps are repeated for each of the sources **code** statements.

The linking step pref. includes creating a procedure execution frame for each procedure defined by the source code statements, each frame having pointers, each of which points to a list of object code blocks having a common property, determing which frame cprresp. to the procedure in which the source code statement belongs, selecting a list of blocks pointed to by a pointer in the determined frame which have a common property with the source **code** statements, and **inserting** the block into the selected list.

ADVANTAGE - In compiling PROLOG programs, allows program statements which use original source code to be compiled.

Dwg.0/5

Title Terms: COMPUTER; SYSTEM; SOURCE; CODE; CREATION; CAPABLE; COMPILE; CODE; INFORMATION; NECESSARY; SOURCE; GENERATE; COMPILE; CODE
Derwent Class: T01
International Patent Class (Main): **G06F-009/45**
International Patent Class (Additional): **G06F-009/44**
File Segment: EPI
?

12/5/1 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO

05247260 **Image available**
CIRCUIT DIAGRAM ERROR CORRECTOR

PUB. NO.: 08-202760 [JP 8202760 A]
PUBLISHED: August 09, 1996 (19960809)
INVENTOR(s): TACHIKAWA YOSHIE
APPLICANT(s): NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 07-031598 [JP 9531598]
FILED: January 27, 1995 (19950127)
INTL CLASS: [6] **G06F-017/50**
JAPIO CLASS: 45.4 (INFORMATION PROCESSING -- Computer Applications)

ABSTRACT

PURPOSE: To simultaneously correct the **error** position of a circuit diagram detected by a check **program** while confirming it on that circuit diagram and to prevent the circuit diagram from being changed regardlessly of the correction of the error position.

CONSTITUTION: An error mark display means 11 displays the graphic data of the circuit diagram to be a **correcting** object **on** a CRT 5a and displays an error mark at the error position shown by respective **error** information outputted from a **check** means 3. When the **error** mark is clicked by a mouse 5c, an error correcting means 13 reads a processing function matched with the error class of the error information corresponding to that error position from a processing function storage means 12 and executes it. Thus, **when that correcting** method is displayed **on** the CRT 5a and a **correction** worker inputs correction data from a keyboard 5b or the like according to that correcting method, the graphic data in a circuit diagram storage means 2 are updated corresponding to those correction data.

12/5/2 (Item 2 from file: 347)
DIALOG(R)File 347:JAPIO

04532309 **Image available**
PROGRAM WRITER FOR IC CARD

PUB. NO.: 06-176209 [JP 6176209 A]
PUBLISHED: June 24, 1994 (19940624)
INVENTOR(s): NAGAMORI AKIO
TAKAHASHI MASASHI
APPLICANT(s): TOPPAN PRINTING CO LTD [000319] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 04-105946 [JP 92105946]
FILED: March 31, 1992 (19920331)
INTL CLASS: [5] G06K-017/00; **G06F-009/445** ; **G06F-009/06**
JAPIO CLASS: 45.3 (INFORMATION PROCESSING -- Input Output Units); 30.1 (MISCELLANEOUS GOODS -- Office Supplies); 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JAPIO KEYWORD: R131 (INFORMATION PROCESSING -- Microcomputers & Microprocessers)
JOURNAL: Section: P, Section No. 1806, Vol. 18, No. 516, Pg. 132, September 28, 1994 (19940928)

ABSTRACT

PURPOSE: To prevent illegal use, the alteration of data, destruction or the runaway of a program by checking a certificate code, which is applied when a terminal is correct, in the case of write so as to judge whether there is any error in translation (compile) or not to download only a correct machine word program.

CONSTITUTION: At a reader/writer 20, a compile means 21 compiles a source

program , and a **check** means 22 **checks** compile **error** and **program** correctness. When the **check** means 22 does not detect any compile **error** or only **when** the source **program** is **correct** based **on** the proper terminal, a code apply means 23 applies the certificate code to an object program. At an IC card 10, a code discriminating means 11 discriminates the presence/absence of the certificate code concerning this object code, and a write means 12 loads down only the correct object program. Thus, illegal use or the like can be prevented.

12/5/3 (Item 3 from file: 347)
DIALOG(R)File 347:JAPIO

04028515 **Image available**
INFORMATION PROCESSOR

PUB. NO.: 05-020215 [JP 5020215 A]
PUBLISHED: January 29, 1993 (19930129)
INVENTOR(s): FUJINO YUKIHIRO
APPLICANT(s): NEC ENG LTD [329822] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 03-176413 [JP 91176413]
FILED: July 17, 1991 (19910717)
INTL CLASS: [5] **G06F-012/16 ; G06F-011/08 ; G06F-011/10 ; G06F-009/34**

JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- Memory Units); 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 1551, Vol. 17, No. 296, Pg. 101, June 07, 1993 (19930607)

ABSTRACT

PURPOSE: To detect a 2-bit error and to detect a failure in an error correcting circuit or the like by executing normal operation at the same time as the case of no error correction.

CONSTITUTION: An output from an **error** correction **code** forming circuit 9 is written in an SPM 4 with the delay of one step from data written from a data register 2 to an SPM 1. An **error** correction **code** is read out from the SPM 4 with the delay of one step from the data read out from the SPM 1. An output from a data register 3 is checked by a parity **checking** circuit 8, and **if** an **error** is present **on** the output, an error **correcting** circuit 7 corrects the error and stores the corrected data in data registers 3, 6, 2. The contents of the register 2 are written in the SPM 1, the formed **error** correction **code** is stored in a data register 5 and then the output of the register 5 is written in the SPM 4 to complete the error correction.

12/5/4 (Item 4 from file: 347)
DIALOG(R)File 347:JAPIO

02074423 **Image available**
ERROR CORRECTING AND DECODING METHOD

PUB. NO.: 61-288523 [JP 61288523 A]
PUBLISHED: December 18, 1986 (19861218)
INVENTOR(s): ISOMURA MASAICHI
APPLICANT(s): MITSUBISHI ELECTRIC CORP [000601] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 60-129504 [JP 85129504]
FILED: June 14, 1985 (19850614)
INTL CLASS: [4] H03M-013/00; **G06F-011/10**
JAPIO CLASS: 42.4 (ELECTRONICS -- Basic Circuits); 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JAPIO KEYWORD: R012 (OPTICAL FIBERS)
JOURNAL: Section: E, Section No. 507, Vol. 11, No. 152, Pg. 51, May

16, 1987 (19870516)

ABSTRACT

PURPOSE: To reduce overlook **error** of instructions by using P decoding to apply **error** correction, **checking** a Q flag when the **error** location is calculated and applying error correction when no Q flag is set and the error location is in the range of 4.

CONSTITUTION: In the decoding method of an **error** correction **code** comprising (m+k+n+1) sets of symbols in total, the 1st decoding is applied at first to (n+1) symbols, and **when** error **correction** is **disabled** in the 1st decoding, a flag is set, and when no flag is set in the 2nd decoding using (m+k+n+1) sets of symbols, error correction is applied only to the (m+k) sets of symbols. When a flag is set, error correction is applied only to (n+1) sets of symbols. Thus, the correction probability of **instructions** and **error** overlook probability being comparatively important in the display data are decreased and the error overlook probability in the decoding is reduced.

12/5/5 (Item 5 from file: 347)

00776857

1-BIT ERROR CORRECTION, 2-BIT ERROR DETECTION SYSTEM

PUB. NO.: 56-097157 [JP 56097157 A]
PUBLISHED: August 05, 1981 (19810805)
INVENTOR(s): NISHIDA HIDEHIKO
APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 54-172326 [JP 79172326]
FILED: December 29, 1979 (19791229)
INTL CLASS: [3] **G06F-011/10**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 86, Vol. 05, No. 169, Pg. 45, October 28, 1981 (19811028)

ABSTRACT

PURPOSE: To prevent the delivery of data in error to the processor, by taking the data in 2-bit error having data in which 1-bit inversion is caused to either of all 0 or all 1 pattern.

CONSTITUTION: The information bit section of 1-bit **error** correction.2-bit error detection SECDED **code** to be **error - checked** , is picked up to form the check bit according to the humming matrix. The exclusive logical sum between the check bit section and that picked up from SEC-DED code, is taken to form the syndrome. If each bit of the syndrome is all 0 or all 1, it is regarded as **correction disable** , and if either 1-bit is 1, the check bit section is formed so **that correction disable** error is included. Thus, the delivery of the data with error to the processor can remarkably be reduced.

12/5/6 (Item 1 from file: 350)

013223791 **Image available**

WPI Acc No: 2000-395665/200034
XRPX Acc No: N00-297419

Memory device has control bus that sends error **correction** code check enable **bit to memory controller so as to activate** error **correction** code **detector for relatively** checking **contents of memory cells in memory array**

Patent Assignee: NEC KYUSHU LTD (KYUN)
Number of Countries: 001 Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 2000137995	A	20000516	JP 98309832	A	19981030	200034	B
JP 3157787	B2	20010416	JP 98309832	A	19981030	200124	

Priority Applications (No Type Date): JP 98309832 A 19981030
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 2000137995	A		11	G11C-029/00	
JP 3157787	B2		8	G11C-029/00	Previous Publ. patent JP 2000137995

Abstract (Basic): JP 2000137995 A

NOVELTY - A memory array (16) has an ECC check bit enable bit memory (14) which stores ECC check enable bit. A control bus (22) connects memory controller (24) and memory (14) and sends ECC check enable bit. A control bus (28), sends ECC check enable bit to a ECC detector (26). A control bus (30) transmits check bit memory selecting signal to ECC detector which enables storing of check bit in check bit memory (12).

DETAILED DESCRIPTION - The check bit memory (12) is connected with ECC **error** detector by **check** bit bus and stores check bit. A data memory (10) connected with error detector by data bus, stores the written data.

USE - Memory device with ECC error correction function.

ADVANTAGE - Provides memory device that enable easy reading of written in data.

DESCRIPTION OF DRAWING(S) - The figure shows the circuit block diagram of ECC error detecting unit of memory device.

Check bit memory (12)
Memory (14)
Memory array (16)
Bus (22)
ECC detector (26)
Control buses (28,30)
pp; 11 DwgNo 2/7

Title Terms: MEMORY; DEVICE; CONTROL; BUS; SEND; ERROR; CORRECT; CODE; CHECK; ENABLE; BIT; MEMORY; CONTROL; SO; ACTIVATE; ERROR; CORRECT; CODE; DETECT; RELATIVELY; CHECK; CONTENT; MEMORY; CELL; MEMORY; ARRAY
Derwent Class: T01; U11; U14
International Patent Class (Main): G11C-029/00
International Patent Class (Additional): **G06F-012/16**
File Segment: EPI

12/5/7 (Item 2 from file: 350)

011010468 **Image available**
WPI Acc No: 1996-507418/199651
XRPX Acc No: N96-427576

Interactive monitor fault **clearing device with built-in** testing - **displays series of** instructions **on screen for user to remove** fault **when no synchronisation signal is received**

Patent Assignee: MAG TECHNOLOGY CO LTD (MAGT-N)
Inventor: CHENG K
Number of Countries: 002 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
DE 29617197	U1	19961114	DE 96U2017197	U	19961002	199651	B
US 5956022	A	19990921	US 96724403	A	19961002	199945	N

Priority Applications (No Type Date): DE 96U2017197 U 19961002; US 96724403 A 19961002
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
DE 29617197	U1		10	G06F-011/00	
US 5956022	A			G09G-005/00	

Abstract (Basic): DE 29617197 U

The device includes a microprocessor control unit. This receives a horizontal synchronisation signal and a vertical synchronisation signal from a video card of a computer system (20). The control unit also displays accurate information about the computer on an image screen. The control unit causes a series of steps for error removal to be displayed on the screen. The steps instruct the user how to clear the problem of not receiving the horizontal and vertical synchronisation signals when the monitor is switched on once but does not receive any horizontal and vertical synchronisation signals.

A horizontal synchronisation signal and a vertical synchronisation signal are received from the video card **while** the monitor is switched **on** . **If** the signals are **correctly** received, information **on** the computer is displayed on the screen. If no synchronisation signal is received when the monitor is on the control unit is used to display the series of steps instructing the user how to remove the error. When the problem is removed information on the computer is displayed on the screen.

ADVANTAGE - Esp. for colour monitors. Provides interactive monitor showing user how to remove faults while computer system is switched on.

Dwg.1/2

Title Terms: INTERACT; MONITOR; FAULT; CLEAR; DEVICE; BUILD; TEST; DISPLAY; SERIES; INSTRUCTION; SCREEN; USER; REMOVE; FAULT; NO; SYNCHRONISATION; SIGNAL; RECEIVE

Derwent Class: P85; T01; T04; W02

International Patent Class (Main): **G06F-011/00** ; G09G-005/00

International Patent Class (Additional): **G06F-003/14** ; H04N-017/04

File Segment: EPI; EngPI

12/5/8 (Item 3 from file: 350)

DIALOG(R)File 350:Derwent WPIX

010724625 **Image available**

WPI Acc No: 1996-221580/199622

XRPX Acc No: N96-186016

Computer error detection and correction system - detects read error , **writes and reads** test **patterns to selected address locations to detect** if **error is fatal, re-loads** correct **copy of data and** enables **computer system, if** error **is not fatal** error , **and aborts** program **if** error **is fatal**

Patent Assignee: UNISYS CORP (BURS)

Inventor: BRUNMEIER T J; BYERS L L; MILLER J A; ROBECK G R

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5511164	A	19960423	US 95396952	A	19950301	199622	B

Priority Applications (No Type Date): US 95396952 A 19950301

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5511164	A		32	G06F-011/00	

Abstract (Basic): US 5511164 A

The source of an error may be a hardware element and the nature of the error may be identified as either fatal or non-fatal. If the nature of the error is considered non-fatal, the error is corrected and the operation of the computer system is continued.

This allows detected errors to be handled immediately after they occur, rather than aborting the operation of the computer system and waiting for a support controller to analyse the **error** , especially important during time critical **operations** .

ADVANTAGE - Identifies source and nature of error, without aborting operation of computer system. This may significantly enhance reliability and performance of corresponding computer system. Since operation of computer system may be aborted in fewer number of times,

amount of data loss is minimised, particularly important for high reliability computer applications, including banking and airline reservation applications, where integrity of database is of utmost importance.

Dwg.2/20

Title Terms: COMPUTER; ERROR; DETECT; CORRECT; SYSTEM; DETECT; READ; ERROR; WRITING; READ; TEST; PATTERN; SELECT; ADDRESS; LOCATE; DETECT; ERROR; FATAL; LOAD; CORRECT; COPY; DATA; ENABLE; COMPUTER; SYSTEM; ERROR; FATAL; ERROR; PROGRAM; ERROR; FATAL

Derwent Class: T01

International Patent Class (Main): **G06F-011/00**

File Segment: EPI

12/5/9 (Item 4 from file: 350)

DIALOG(R)File 350:Derwent WPIX

007486805

WPI Acc No: 1988-120738/198818

XRPX Acc No: N88-091653

PCM signal reproduction appts. with error correction - has two decoders and two prepn. point detectors to maximise correction during reproduction from magnetic tape

Patent Assignee: MITSUBISHI DENKI KK (MITQ)

Inventor: INOUE T; ONISHI K; SUGIYAMA K

Number of Countries: 003 Number of Patents: 005

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
DE 3735979	A	19880428	DE 3735979	A	19871023	198818	B
GB 2197509	A	19880518	GB 8724878	A	19871023	198820	
US 4829525	A	19890509	US 87112035	A	19871023	198922	
GB 2197509	B	19910626				199126	
DE 3735979	C	19911219				199151	

Priority Applications (No Type Date): JP 87149448 A 19870615; JP 86253337 A 19861024; JP 87149446 A 19870615; JP 87149447 A 19870615

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
DE 3735979	A		34		
US 4829525	A		33		

Abstract (Basic): DE 3735979 A

A head reads digital data on a multi-track tape contg. main data, an **error** detection code and an **error** correction **code** . A **test** circuit **checks** the sensed **error** detection **code** and sends a **test** signal to a prepn. point detector. A decoder with correction facility decodes the digital data from the test circuit so **that** the main data are **corrected on** the basis of the **correction** code. A second decoder with detection facility decodes the data from the **test** circuit to detect the **error** detection **code** .

A second prepn. point detector produces a signal when the second decoder detects the **error** detection **code** in the data. A prepn. circuit prepared the data when at least one of the prepn. point signals is generated, and selectors energies the first or second decoder on the basis of the first prepn. point signal.

5/18

Title Terms: PCM; SIGNAL; REPRODUCE; APPARATUS; ERROR; CORRECT; TWO; DECODE ; TWO; PREPARATION; POINT; DETECT; MAXIMISE; CORRECT; REPRODUCE; MAGNETIC ; TAPE

Derwent Class: T03

International Patent Class (Additional): **G06F-011/10** ; G11B-005/09; G11B-020/12

File Segment: EPI

12/5/10 (Item 5 from file: 350)

DIALOG(R)File 350:Derwent WPIX

004585887
WPI Acc No: 1986-089231/198614
XRPX Acc No: N86-065265

Digital data error correction system for memory - has parity bit and code generators deriving two codes to provide self-checking facility

Patent Assignee: NEC CORP (NIDE)
Inventor: KIMURA T; MASUHARA H

Number of Countries: 004 Number of Patents: 004
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 176218	A	19860402				198614	B
US 4716566	A	19871229	US 85767596	A	19850820	198802	
EP 176218	B	19901017				199042	
DE 3580147	G	19901122				199048	

Priority Applications (No Type Date): JP 84172670 A 19840820
Cited Patents: 2.Jnl.Ref; A3...8743; JP 58101539; JP 59131237; No-SR.Pub; US 3800281
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 176218	A	E	23		
Designated States (Regional): DE FR GB					
EP 176218	B				
Designated States (Regional): DE FR GB					

Abstract (Basic): EP 176218 B

A **code** generator (2) outputs an **error** correcting **code** (C) which is added to data (X) to be transmitted. A parity bit (p) for the data is supplied by a generator (3) to an output buffer (4) which receives the **error** correcting **code** and data and passes them all to an external memory (5).

An input buffer (6) receives data (Xa) with a received **error** -correcting **code** (Ca) and parity bit (Pa) which correspond to the data, passed to the memory.

A second **code** generator (7) and parity bit generator provide a respective second **error** -correcting **code** (Cb) and parity bit (Pb) for addition to the received data. A signal generator (9) produces an error correcting signal (s) if it determines a difference between the **error** correcting codes. A **checker** (10) outputs an **error** signal (m) if the two parity bits do not agree. An enabling circuit (11) provides in response to the error correcting signal and parity error signal, an enabling signal (n) to a circuit (12) which then corrects the error of the received data (Xa).

ADVANTAGE - **Correction** does not take place **when** **enabling** signal is absent. (23pp Dwg.No.1/4)

Title Terms: DIGITAL; DATA; ERROR; CORRECT; SYSTEM; MEMORY; PARITY; BIT; CODE; GENERATOR; DERIVATIVE; TWO; CODE; SELF; CHECK; FACILITY
Index Terms/Additional Words: RECORD
Derwent Class: T03; U21; W01
International Patent Class (Additional): **G06F-011/10** ; **G06F-012/16** ; G11B-020/18; H03M-013/00
File Segment: EPI
?

14/5/2 (Item 2 from file: 347)
DIALOG(R)File 347:JAPIO

07155632 **Image available**
SOFTWARE DEVELOPMENT SUPPORT SYSTEM

PUB. NO.: 2002-024013 [JP 2002024013 A]
PUBLISHED: January 25, 2002 (20020125)
INVENTOR(s): OKUMOTO FUMIHIRO
YAMAZAKI MICHIHIRO
TOMITA OSAMU
AKAHA KOICHI
APPLICANT(s): HITACHI LTD
HITACHI SOFTWARE ENG CO LTD
APPL. NO.: 2000-204590 [JP 2000204590]
FILED: July 06, 2000 (20000706)
INTL CLASS: **G06F-009/44**

ABSTRACT
PROBLEM TO BE SOLVED: To automatically adjust versions consistency in the case of compositing softwares and also to promptly deal with **fault** occurrence when the operation confirmation **test** of the composite **software** is performed.
SOLUTION: Software parts developed at development points A 210 and B 220, etc., are collected to a software centralized control center 230 by e-mail, etc., the consistency to management information in a database 232 is confirmed about the person in charge of development of each part, version, etc., and stored in the database, when the whole parts are collected, the whole parts are copied into a test machine 232, operation confirmation is performed as one piece of **software** composed of the whole parts, when a **failure** takes **place** in a part during the test, a **failure** automatic reporting **program** 236 automatically detects the **failure** , transmits an **error code** , **error** contents, etc., to a corresponding point by e-mail, etc., and when there is no **abnormal state** in the whole parts, a medium preparing machine 234 prepares software in a distributable form, and a completion automatic reporting program 237 transmits completion report notification to each point by e-mail, etc.

14/5/3 (Item 3 from file: 347)
DIALOG(R)File 347:JAPIO

04533964 **Image available**
FAULT DETECTING CIRCUIT

PUB. NO.: 06-177864 [JP 6177864 A]
PUBLISHED: June 24, 1994 (19940624)
INVENTOR(s): TADOKORO NAOAKI
APPLICANT(s): NEC COMMUN SYST LTD [491066] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 04-321712 [JP 92321712]
FILED: December 01, 1992 (19921201)
INTL CLASS: [5] H04L-001/20; **G06F-011/10** ; H04L-001/00; H04L-025/03
JAPIO CLASS: 44.3 (COMMUNICATION -- Telegraphy); 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: E, Section No. 1610, Vol. 18, No. 509, Pg. 121, September 26, 1994 (19940926)

ABSTRACT
PURPOSE: To realize the discrimination of the intermittent fixed **fault** and normality being a specific **state** transition by the circuit of a hardware by detecting the occurrence of a fault in communication information at the reception side of a communication line which transmits fixed length information from the arithmetic result of a **fault** detection

code added to the fixed length information.

CONSTITUTION: This circuit is provided with a means which receives and decodes a data signal including an **error check code** for detecting the **error** of the communication information of the fixed length, and a means which operates an error **correction** encoding based **on** the information of the decoded **error** check **code** , and outputs an **abnormal** or normal **state** signal. The circuit is provided with a first counting means 13 which inputs the **abnormal state** signal and counts the number of times of the abnormality, a second counting means 19 which inputs the normal state signal and counts the number of times of the normality, a discrimination value setting means 11 which sets a threshold value for the count value of the first counting means 13, and a means 16 which resets the second counting means 19 when the **abnormal state** signal is inputted.

14/5/5 (Item 5 from file: 347)
DIALOG(R)File 347:JAPIO

03482324 **Image available**
ERROR CORRECTION DEVICE

PUB. NO.: 03-145224 [JP 3145224 A]
PUBLISHED: June 20, 1991 (19910620)
INVENTOR(s): ARAI MASAKI
APPLICANT(s): MATSUSHITA GRAPHIC COMMUN SYST INC [330729] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 01-283285 [JP 89283285]
FILED: October 30, 1989 (19891030)
INTL CLASS: [5] H03M-013/00; **G06F-011/10**
JAPIO CLASS: 42.4 (ELECTRONICS -- Basic Circuits); 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: E, Section No. 1112, Vol. 15, No. 367, Pg. 68, September 17, 1991 (19910917)

ABSTRACT

PURPOSE: To improve the **correction** capability of an error caused **on** a transmission line without much deteriorating the transmission efficiency by correcting the detected error in any axial direction of 3-dimension matrix based **on** an error **correction** parity data.

CONSTITUTION: An **error** correction coding section 100 **adding** an **error** correction redundancy **code** to a data transmission side and an **error** correction decoding section 200 detecting and correcting an error to a data reception side are provided. Then a frame in the unit of data transmission is formed to be a 3-dimension matrix at the sender side and an **error** detection **check** data is added to a data in the frame and a parity data for error correction is added in the 3-dimensional direction of the frame. A data reception side detects a data error sent in the unit of frames by a **check** data and the detected **error** is corrected from any axial direction of 3-dimension matrix based **on** the error **correction** parity data. Thus, the **correction** capability of an error caused **on** a transmission line L without much deteriorating the transmission efficiency is enhanced.

14/5/7 (Item 7 from file: 347)
DIALOG(R)File 347:JAPIO

02904445 **Image available**
FAULT SUPERVISORY SYSTEM

PUB. NO.: 01-202045 [JP 1202045 A]
PUBLISHED: August 15, 1989 (19890815)
INVENTOR(s): BEPPU YUICHIRO
APPLICANT(s): NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)

APPL. NO.: 63-025719 [8825719]
FILED: February 08, 1988 (19880208)
INTL CLASS: [4] H04L-001/00; **G06F-011/10** ; H03M-013/00
JAPIO CLASS: 44.3 (COMMUNICATION -- Telegraphy); 42.4 (ELECTRONICS -- Basic Circuits); 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: E, Section No. 844, Vol. 13, No. 503, Pg. 101, November 13, 1989 (19891113)

ABSTRACT

PURPOSE: To make it possible to supervise a fault, as well at the time of inputting a specific input signal by **adding** supervising function of a storage circuit **fault** using a circulating **code** to supervise the **fault** of signal based upon parity **check** .

CONSTITUTION: The signal parity bits of input signal lines 1-3 are respectively counted by parity counting circuits 4-6 and respective counted results are sent to a multiplexing circuit 8. The circuit 8 multiplexes a cyclic code generated from an input side cyclic code generating circuit 7 and the parity counting results sent from the circuits 4-6 **on** a certain **fixed** frame format. A parity correction deciding circuit 15 executes parity check and a circulating code correction deciding circuit 16 decides coincidence or discordance between the circulating signal generated from the circuit 7 and extracted by an extracting circuit 13 and a circulating code generated from an output side circulating **code** generating circuit 14 to execute **error check** to be decided as correct at the time of coincidence or as error at the time of discordance. Even if a specific input signal is inputted at the time of data transmission, the generation of the condition that the supervision for a storage circuit is impossible can be prevented.

14/5/8 (Item 8 from file: 347)

02670045 **Image available**
DATA PROCESSOR

PUB. NO.: 63-286945 [JP 63286945 A]
PUBLISHED: November 24, 1988 (19881124)
INVENTOR(s): ONODA TAKASHI
APPLICANT(s): HITACHI LTD [000510] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 62-121211 [JP 87121211]
FILED: May 20, 1987 (19870520)
INTL CLASS: [4] **G06F-012/14**
JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- Memory Units); 29.4 (PRECISION INSTRUMENTS -- Business Machines)
JOURNAL: Section: P, Section No. 843, Vol. 13, No. 108, Pg. 163, March 15, 1989 (19890315)

ABSTRACT

PURPOSE: To make it detectable that a **check error** has occurred in an area **checking** means and a data is nor secured by using a fact that a check code is not updated while an unjustifiable data rewrite is performed.
CONSTITUTION: A transaction processing part 6 controls a storage to a storage part 4 so as always to be performed in a check code update mode through a rewrite control part 7. The control part 7 stores a given data in the storage part 4, and besides, updates the corresponding check **code** as well, when the check **code** update mode is instructed and if a **place** where the data is stored is a data security area. When the processing part 6 rewrites the data security area 9 of the storage part 4 justly, the check code is always and **correctly** updated as well. **On** the other hand, when the data security area 9 is rewritten unjustly, because the check **code** is not updated, the **check error** is detected by an area **checking** part 8. An area check is performed if necessary, and the contents of the **check error** is displayed on a display part 3 together with the name of a

generating area through the transaction processing part 6.

14/5/9 (Item 9 from file: 347)
DIALOG(R)File 347:JAPIO

02566030 **Image available**
COMMUNICATION ERROR CORRECTION SYSTEM FOR TIME INFORMATION

PUB. NO.: 63-182930 [JP 63182930 A]
PUBLISHED: July 28, 1988 (19880728)
INVENTOR(s): NAGASAWA MASASHI
APPLICANT(s): NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 62-013798 [JP 8713798]
FILED: January 23, 1987 (19870123)
INTL CLASS: [4] H04L-001/00; **G06F-013/00** ; H04L-007/00
JAPIO CLASS: 44.3 (COMMUNICATION -- Telegraphy); 45.2 (INFORMATION PROCESSING -- Memory Units)
JOURNAL: Section: E, Section No. 688, Vol. 12, No. 457, Pg. 150, November 30, 1988 (19881130)

ABSTRACT

PURPOSE: To easily correct the communication **error** of time information by sending a time data, a CRC **code** , a strobe signal and a synchronizing signal from a timer and **adding** a time of 1sec to a time of reception just before a received information processing unit detects an error through the adoption of the constitution.

CONSTITUTION: A reception register 45 is a shift register of serial input/parallel output and shifts in a time data signal 32 and a strobe signal 33 synchronously. A reception register 45 has a length shifting in the time information and the CRC code and only the time information is outputted to the output line 50. In case of receiving the time data, a CRC **check** circuit 46 detects an **error** by using a time data and a succeeding CRC **code** and gives an **error** detection signal 47 to a selection circuit 44. The selection circuit 44 selects an output 50 of the reception register when the error detection signal 47 is not in error and selects an output 49 of a 1sec adder circuit 43 outputting the addition of 1sec to the content of the time register 42 in case of the **error state** .

14/5/10 (Item 10 from file: 347)
DIALOG(R)File 347:JAPIO

00790897 **Image available**
TWO-BIT ERROR CORRECTION SYSTEM

PUB. NO.: 56-111197 [JP 56111197 A]
PUBLISHED: September 02, 1981 (19810902)
INVENTOR(s): AOKI TAKASHI
APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 55-011328 [JP 8011328]
FILED: February 01, 1980 (19800201)
INTL CLASS: [3] G11C-029/00; **G06F-011/08**
JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- Memory Units); 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 90, Vol. 05, No. 182, Pg. 143, November 20, 1981 (19811120)

ABSTRACT

PURPOSE: To **enable** to **correct** 2-bit **errors** without using the two-bit **error** correcting **code** (ECC), by the parity check per each block of a plurality of data **added** with the 2-bit **error** detection **code** for **error** correction.

CONSTITUTION: A plurality of data added with ECC having 1-bit error correction 2-bit error detecting functions in which one word has a given bit, are stored in the memory 1. At the same time, parity bits are provided every bit location of each word and this is stored in the block parity area 1-2 of the memory 1. With this state, the data such as address YA of the memory 1 are read out and a bit **error** is detected with ECC of the **error** correcting **code** area, then **checking** is made with the parity data of the area 1-2 for each data block per bit, and the error production bits dm, dn are detected from the error presence bits Xm, Xn for the address YA, allowing to correct the 2-bit error with a simple constitution not requiring 2-bit the correction ECC.

14/5/12 (Item 12 from file: 347)
DIALOG(R)File 347:JAPIO

00507141
ERROR INSPECTION AND CORRECTION CIRCUIT

PUB. NO.: 54-159141 [JP 54159141 A]
PUBLISHED: December 15, 1979 (19791215)
INVENTOR(s): TAKAHASHI YUKIO
HAGIWARA NOBORU
KOBAYASHI HIDEHIKO
APPLICANT(s): NIPPON TELEGR & TELEPH CORP <NTT> [000422] (A Japanese Company or Corporation), JP (Japan)
NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 53-068669 [JP 7868669]
FILED: June 06, 1978 (19780606)
INTL CLASS: [2] **G06F-011/10** ; G11C-029/00
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units); 45.2 (INFORMATION PROCESSING -- Memory Units)
JOURNAL: Section: E, Section No. 171, Vol. 04, No. 18, Pg. 159, February 13, 1980 (19800213)

ABSTRACT

PURPOSE: To inspect the error and correct it to arbitray length of information data, by inputting the information bit changing the element of one row and column selected with the input information and the parity **check** matrix HMt to the **error** correction **code** generation circuit.

CONSTITUTION: The input information data 1 corresponding to the row of weight in an even number **added** with one element based **on** HMt expressing the **error correction code** is inputted to the **error** correction **code** /syndrome generation circuit 6. The information bit 2 **adding** the element of 1 to the row j selected with HMt is inputted to the selection circuit 4, and the information bit 5 is outputted, equal to 0 or bit 2, to input it to the circuit 6 depending on the state of 0 or 1 of the selection signal 3. The circuit 6 outputs the **error** correction **code** bit or the syndrome bit 7 based on HMt. Further, the syndrome 8 enters the syndrome decoders 9 and 10 corresponding to the row j based on HMt, they are inputted to the selection circuit 13 with the first and second correction position designation signals 11 and 12, and the signal 11 or 12 is selected depending on the state of the selection signal 3 and outputted.

14/5/13 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX

012310023 **Image available**
WPI Acc No: 1999-116129/199910
XRPX Acc No: N99-085718
Run time error testing **system for `C' language** program **- appends test codes before and after condition judgment statement, and displays**

dialog box according to [redacted]st code generated during execu[redacted]n
Patent Assignee: NEC SOFTWARE SHOKOKU LTD (NIDE)
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 10340207	A	19981222	JP 97148977	A	19970606	199910	B

Priority Applications (No Type Date): JP 97148977 A 19970606
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 10340207	A		7	G06F-011/28	

Abstract (Basic): JP 10340207 A
NOVELTY - A test **code** is **appended** before and after a condition judgment statement of a source program checked by analyser. A dialog box is displayed during execution of the condition statement indicating the list code about the variable used in the condition statement.
USE - For detecting **errors** in **condition** statements such as `if', `for', `while', `switch' statements in C-language.
ADVANTAGE - The value of the variable used for condition judgment program is confirmed during execution, since the dialog box is displayed interrupting the process during execution of conditional statements. Test data is collected with the source program in applicable location. The number of processes for confirming the values of all instructions in source program is reduced. Automatically resets the value of variable used for condition judgment statement.
Dwg.1/9
Title Terms: RUN; TIME; ERROR; TEST; SYSTEM; LANGUAGE; PROGRAM; TEST; CODE; AFTER; CONDITION; STATEMENT; DISPLAY; BOX; ACCORD; TEST; CODE; GENERATE; EXECUTE
Derwent Class: T01
International Patent Class (Main): **G06F-011/28**
File Segment: EPI

14/5/14 (Item 2 from file: 350)
DIALOG(R)File 350:Derwent WPIX

012184310 **Image available**
WPI Acc No: 1998-601223/199851
XRPX Acc No: N98-468613
Forwarding data check apparatus - checks **forwarding** error **in actual data obtained after removing additional data included in received data**
Patent Assignee: NEC KOFU LTD (NIDE)
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 10271097	A	19981009	JP 9769742	A	19970324	199851	B

Priority Applications (No Type Date): JP 9769742 A 19970324
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 10271097	A		5	H04L-001/00	

Abstract (Basic): JP 10271097 A
The apparatus has a transmitter (10) which forward fixed length data to a receiver (20). In the transmitter, the transmitting data is generated by adding arbitrary fixed values to the actual data to be transmitted. Then a check **code** is **added** to the transmitted data and transmitted to the receiver.
The receiver then obtains the actual data removing the additional data included in the data received from the transmitter and then checks the actual data by using the check **code** in order to detect any forwarding **error** . Then the actual data are sent out of the receiver.
USE - In information processing system.
ADVANTAGE - **Checks error** in actual data, efficiently. **Enables** to **restore** failure efficiently.

Dwg.1/3
Title Terms: FORWARDING; DATA; CHECK; APPARATUS; CHECK; FORWARDING; ERROR; ACTUAL; DATA; OBTAIN; AFTER; REMOVE; ADD; DATA; RECEIVE; DATA
Derwent Class: T01; U21; W01
International Patent Class (Main): H04L-001/00
International Patent Class (Additional): **G06F-011/10**
File Segment: EPI

14/5/16 (Item 4 from file: 350)
DIALOG(R)File 350:Derwent WPIX

009467030 **Image available**
WPI Acc No: 1993-160569/199320
XRPX Acc No: N93-123253
Fault indication in array data storage system - providing code **relating to data transfer** fault **and accessing** code **to take** corrective **action based** on code **information.**
Patent Assignee: FUJITSU LTD (FUIT)
Inventor: MORGAN L A; PARRISH M
Number of Countries: 004 Number of Patents: 006
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 541991	A2	19930519	EP 92117804	A	19921019	199320	B
US 5379411	A	19950103	US 91792702	A	19911115	199507	
EP 541991	A3	19940216	EP 92117804	A	19921019	199518	
US 5574856	A	19961112	US 91792702	A	19911115	199651	
			US 94366032	A	19941229		
EP 541991	B1	19970625	EP 92117804	A	19921019	199730	
DE 69220553	E	19970731	DE 620553	A	19921019	199736	
			EP 92117804	A	19921019		

Priority Applications (No Type Date): US 91792702 A 19911115; US 94366032 A 19941229
Cited Patents: No-SR.Pub; EP 416968; EP 442531; US 4598357; WO 9113405
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 541991	A2	E	15		
Designated States (Regional): DE FR GB					
US 5379411	A		14		
US 5574856	A		14		Cont of application US 91792702
					Cont of patent US 5379411
EP 541991	B1	E	16		
Designated States (Regional): DE FR GB					
DE 69220553	E				Based on patent EP 541991

Abstract (Basic): EP 541991 A
The method involves **appending** a **code** byte, with a number of **code** bits, to data stored in the array storage system (14) such that, when a **fault** occurs, predetermined **code** bits are set to indicate the data operation that was taking **place** when the **fault** occurred. This **code** can then be accessed in order to take **corrective** action based **on** the code bits set.
Code bits can be set for two data transfer related operations; a data reconstruction operation and a data reassignment operation. In the data reconstruction operation a determination is made that a storage device cannot be properly used to reconstruct data. In the data reassignment operation the data is written to an alternate storage location in which the storing of data was not successfully completed.
ADVANTAGE - Enables operation **error** identification without use of **check** bits.
Dwg.1/4
Title Terms: FAULT; INDICATE; ARRAY; DATA; STORAGE; SYSTEM; CODE; RELATED; DATA; TRANSFER; FAULT; ACCESS; CODE; CORRECT; ACTION; BASED; CODE; INFORMATION
Derwent Class: T01
International Patent Class (Main): **G06F-011/00** ; **G06F-011/08**

File Segment: EPI

14/5/17 (Item 5 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2003 Thomson Derwent. All rts. reserv.

009290562 **Image available**
WPI Acc No: 1992-417971/199251
XRPX Acc No: N92-318755
Modifying program code in radiotelephone - by loading programme code via bus from external device and storing code in RAM
Patent Assignee: NOKIA MATKAPUHELIMET OY (OYNO); NOKIA MOBILE PHONES LTD (OYNO)
Inventor: PELKONEN S; PERLKONEN S
Number of Countries: 003 Number of Patents: 004
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
GB 2256734	A	19921216	GB 9212197	A	19920609	199251	B
FI 9102875	A	19921215	FI 912875	A	19910614	199310	
US 5349697	A	19940920	US 92897280	A	19920611	199437	
GB 2256734	B	19950830	GB 9212197	A	19920609	199538	

Priority Applications (No Type Date): FI 912875 A 19910614
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
GB 2256734	A		9	G06F-009/24	
US 5349697	A		4	H04B-001/38	
GB 2256734	B		1	G06F-009/24	
FI 9102875	A			H04B-001/16	

Abstract (Basic): GB 2256734 A
The programme code of a radiotelephone is stored in a R/W RAM (3) rather than in an EPROM as is conventional. The checking of the programme code and the loading of the programme from an external device coupled thereto via an external bus is controlled by programmes recorded in a ROM (4).
When an error is detected the external device is coupled to the telephone, and upon receipt of a pre-determined instruction from the external bus, programme code is supplied from the external device to the RAM for storage. The subsequently stored programme **code** is re-**checked** to detect any **error** in the stored programme **code** and whereby the logic is **placed** in a waiting **state** upon detection of an **error** until an **error** free programme **code** is recorded into the RAM.
USE/ADVANTAGE - In allows programme code to be changed and updated as required without having to physically replace memory device, eg. the EPROM mobile and cellular telephones.
Dwg.1/1
Title Terms: MODIFIED; PROGRAM; CODE; RADIOTELEPHONE; LOAD; PROGRAMME; CODE ; BUS; EXTERNAL; DEVICE; STORAGE; CODE; RAM
Derwent Class: T01; W01
International Patent Class (Main): **G06F-009/24** ; H04B-001/16; H04B-001/38
International Patent Class (Additional): H04B-017/00
File Segment: EPI

14/5/29 (Item 17 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2003 Thomson Derwent. All rts. reserv.

002126474
WPI Acc No: 1979-E6405B/197921
Error correcting microinstruction retriever for digital computer - has output from microprogram control to error **detector and data receiver for** test **digits register**
Patent Assignee: GUSHENSKOV B N (GUSH-I)
Inventor: SAMARSKII A S; VOLKOVA N A

Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
SU 615478	A	19780616				197921 B

Priority Applications (No Type Date): SU 2387616 A 19760723

Abstract (Basic): SU 615478 A

Prior circuitry contains microprogram control (1), memory (2), results recorder (3), data register (4), test digits register (5) and generator (6), comparator (7), syndrome register (8), decoder (9) and corrector (10). The error pre-detector (11) and data receiver (12) are introduced in order to increase operating speed.

This microinstruction retriever is useful in electronic digital computing machines. A microinstruction is normally produced with no loss of time greater than memory readout time. But error-affected microinstructions are repeated with corrected codes.

The presence of **error** is determined by the '1' **state** of at least one digit **place** of a syndrome **code** . The **error** signal to the results recorder leads to blocking of the next microinstruction from the memory. In exchange with the memory the decoder determines the type of error, whether correctible or incorrectible. The correction signals invert the faulty digit.

Title Terms: ERROR; CORRECT; MICROINSTRUCTION; RETRIEVAL; DIGITAL; COMPUTER ; OUTPUT; MICROPROGRAM; CONTROL; ERROR; DETECT; DATA; RECEIVE; TEST; DIGITAL; REGISTER

Derwent Class: T01

International Patent Class (Additional): **G06F-009/14**

File Segment: EPI

```
File 348:EUROPEAN PATENTS 1978-2003/Jun W01
         (c) 2003 European Patent Office
File 349:PCT FULLTEXT 1979-2002/UB=20030605,UT=20030529
         (c) 2003 WIPO/Univentio

Set     Items   Description
S1       6489   ASSERTION? ? OR ASSERT
S2      39669   (ERROR? ? OR FAIL? OR FAULT? ? OR PROBLEM? ? OR TROUBLE OR
             FLAW OR IRREGULAR? OR GLITCH?? OR FALSE OR ODD???? OR STRANGE-
             ???? OR ANOMAL? OR ABNORMAL?)(5N)(TEST??? OR CHECK???)
S3      61761   (ERROR? ? OR FAIL? OR FAULT? ? OR PROBLEM? ? OR TROUBLE OR
             FLAW OR IRREGULAR? OR GLITCH?? OR FALSE OR ODD???? OR STRANGE-
             ???? OR ANOMAL? OR ABNORMAL?)(5N)(CONDITION? ? OR STATE OR ST-
             ATES OR SITUATION OR STATUS)
S4     247086   (RECOVER? OR CORRECT? OR FIX??? OR MEND??? OR REMED??? OR -
             RECTIF? OR REPAIR? OR PATCH? OR RESTOR? OR RESOLV? OR SOLV?)(-
             5N)(ENABL? OR ON OR DISABL? OR OFF)
S5      29556   (INSERT? OR PUT???? OR PLAC??? OR PLACEMENT OR ADD??? OR A-
             PPEND?)(5N)(CODE? ? OR INSTRUCTION? ? OR FUNCTION? ? OR COMMA-
             ND? ? OR ROUTINE? ? OR PROCEDURE? ?)(5N)(PROGRAM? ? OR CODE OR
              APPLICATION? ? OR SOFTWARE)
S6      86338   (ERROR? ? OR FAIL? OR FAULT? ? OR PROBLEM? ? OR TROUBLE OR
             FLAW OR IRREGULAR? OR GLITCH?? OR FALSE OR ODD???? OR STRANGE-
             ???? OR ANOMAL? OR ABNORMAL?)(10N)(PROGRAM? ? OR CODE OR APPL-
             ICATION? ? OR SOFTWARE OR INSTRUCTIONS OR OPERATIONS)
S7         53   S1(S)S4:S5(S)S6 AND IC=G06F
S8        202   S2(S)S4(S)S6 AND IC=G06F
S9      27514   (WHEN OR IF OR SHOULD OR WHILE OR THAT)(5W)S4
S10        78   S2(S)S9(S)S6 AND IC=G06F
S11        62   S10 NOT S7
S12       232   S2(S)S5(S)S6 AND IC=G06F
S13     86659   (INSERT? OR PUT???? OR PLAC??? OR PLACEMENT OR ADD??? OR A-
             PPEND?)(5N)(CODE? ? OR INSTRUCTION? ? OR FUNCTION? ? OR COMMA-
             ND? ? OR ROUTINE? ? OR PROCEDURE? ?)
S14     18122   S13(5W)(PROGRAM? ? OR CODE OR APPLICATION? ? OR SOFTWARE)
S15       150   S2(S)S14(S)S6 AND IC=G06F
S16       116   S15 NOT (S7 OR S11)
S17        19   S16/TI,AB,CM
S18        94   S2(S)S3:S4(S)S13(S)S6 AND IC=G06F
S19        52   S18 NOT (S7 OR S11)
S20       330   S3(S)S4:S5(S)S6 AND IC=G06F
S21       200   S3(S)S4(S)S6 AND IC=G06F
S22        81   S3(S)S9(S)S6 AND IC=G06F
S23        29   S22 NOT (S7 OR S11 OR S19)
S24       186   S3(S)S5(S)S6 AND IC=G06F
S25        72   S3(S)(S2 OR S4)(S)S5(S)S6 AND IC=G06F
S26        14   S25 NOT (S7 OR S11 OR S19 OR S23)
S27        61   S2(S)S3(S)S4(S)S5
S28        37   S27 AND IC=G06F
S29         1   S28 NOT (S7 OR S11 OR S19 OR S23 OR S26)
```

7/5,K/2 (Item 2 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00897682
Fault isolation
Fehlereingrenzung
Localisation de fautes
PATENT ASSIGNEE:
Compaq Computer Corporation, (687792), 20555 S.H. 249, Houston Texas 77070, (US), (Proprietor designated states: all)
INVENTOR:
Goodrum, Alan L., 16522 Avenfield, Tomball, Texas 77375, (US)
Autor, Jeffrey S., 12514 Logan Mill, Houston, Texas 77070, (US)
Culley, Paul R., 13010 Marron Drive, Cypress, Texas 77429, (US)
Miller, Joseph P., 12906 Golden Rainbow Drive, Cypress, Texas 77429, (US)
Tavallaei, Siamak, 9418 Landry Boulevard, Spring, Texas 77379, (US)
Basile, Barry S., 18314 Elmdon, Houston, Texas 77084, (US)
Richard, Elizabeth A., 11800 Grant Road, No. 2303, Cypress, Texas 77429, (US)
Rose, Eric E., 7010 Huntbrook Drive, Spring, Texas 77379, (US)
LEGAL REPRESENTATIVE:
Brunner, Michael John et al (28871), GILL JENNINGS & EVERY, Broadgate House, 7 Eldon Street, London EC2M 7LH, (GB)
PATENT (CC, No, Kind, Date): EP 820012 A2 980121 (Basic)
EP 820012 A3 990113
EP 820012 B1 030507
APPLICATION (CC, No, Date): EP 97303790 970604;
PRIORITY (CC, No, Date): US 658750 960605
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: **G06F-011/14**
CITED PATENTS (EP B): GB 2292238 A
CITED REFERENCES (EP B):
"Isolating the SOurce of Small COmputer System Interface Bus Hang Error at Run-Time" IBM TECHNICAL DISCLOSURE BULLETIN., vol. 39, no. 8, August 1996, page 61 XP000638138 NEW YORK US;

ABSTRACT EP 820012 A2
A device causing a faulty condition in a computer system having devices is isolated by detecting for a faulty condition associated with the devices and identifying the device causing the faulty condition. The devices are coupled to a bus. The faulty condition includes a bus hang condition. The devices are turned off when a bus hang condition is detected. The devices are then turned back on to test the devices. Each device is tested by writing and reading its configuration space. Information on the bus associated with the faulty condition is stored. The stored information is retrieved after the faulty condition has occurred, with the stored information including address, data, and bus control information.
ABSTRACT WORD COUNT: 115
NOTE:
Figure number on first page: 40

LEGAL STATUS (Type, Pub Date, Kind, Text):

Examination:	011010 A2	Date of dispatch of the first examination report: 20010828
Application:	980121 A2	Published application (Alwith Search Report ;A2without Search Report)
Grant:	030507 B1	Granted patent
Search Report:	990113 A3	Separate publication of the European or International search report
Change:	990120 A2	International patent classification (change)
Examination:	990901 A2	Date of request for examination: 19990705

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	199804	968
CLAIMS B	(English)	200319	675

CLAIMS B (German) 200319 667
CLAIMS B (French) 200319 798
SPEC A (English) 199804 59653
SPEC B (English) 200319 59722
Total word count - document A 60629
Total word count - document B 61862
Total word count - documents A + B 122491

INTERNATIONAL PATENT CLASS: **G06F-011/14**

...SPECIFICATION active). The bus hang recovery state machine 456 ensures that the secondary PCI bus 32 is brought back to the idle state to allow the **software** to isolate the **faulty** slot. When the hang condition is detected, the SIO 50 powers down the secondary bus slots, which would automatically place the bus 32 into the...

...chip 48 was the master) when the bus hang occurred, then the bridge chip 48 would remain on the bus. To take the bridge chip **off** the bus, the **recovery** state machine 456 forces a retry cycle on the PCI bus 32 by asserting the signal STOP(underscore).
A bus history capture block 458 monitors...

...SPECIFICATION active). The bus hang recovery state machine 456 ensures that the secondary PCI bus 32 is brought back to the idle state to allow the **software** to isolate the **faulty** slot. When the hang condition is detected, the SIO 50 powers down the secondary bus slots, which would automatically place the bus 32 into the...

...chip 48 was the master) when the bus hang occurred, then the bridge chip 48 would remain on the bus. To take the bridge chip **off** the bus, the **recovery** state machine 456 forces a retry cycle on the PCI bus 32 by asserting the signal STOP(underscore).
A bus history capture block 458 monitors...

7/5,K/10 (Item 10 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00887645
Management of overflowing data in a computer system
Datenuberlaufverwaltung in einem Rechnersystem
Gestion de debordement de donnees dans un systeme d'ordinateur
PATENT ASSIGNEE:
Compaq Computer Corporation, (687792), 20555 S.H. 249, Houston Texas 77070, (US), (applicant designated states: AT;BE;CH;DE;DK;ES;FI;FR;GB;GR;IE;IT;LI;LU;MC;NL;PT;SE)
INVENTOR:
Pettey, Christopher J., 9327 Rodgers Road, Houston, Texas 77070, (US)
Maclaren, John M., 15318 Redbud Leaf Lane, Cypress, Texas 77429, (US)
LEGAL REPRESENTATIVE:
Brunner, Michael John et al (28871), GILL JENNINGS & EVERY Broadgate House 7 Eldon Street, London EC2M 7LH, (GB)
PATENT (CC, No, Kind, Date): EP 811934 A2 971210 (Basic)
EP 811934 A3 990210
APPLICATION (CC, No, Date): EP 97303804 970604;
PRIORITY (CC, No, Date): US 658533 960605
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: **G06F-013/40**
ABSTRACT EP 811934 A2
A computer system includes a data storage device on the first data bus, a device on the second data bus, and a bridge device capable of storing multiple data transactions for delivery from the second data bus to the first data bus. The bridge device includes a first data storage buffer preassigned to one of the transactions held in the bridge, and a buffer management element that assigns, if necessary, a second data storage buffer to the data transaction when the data associated with the transaction overflows the first data storage buffer.

ABSTRACT WORD COUNT: 93

LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: 971210 A2 Published application (Alwith Search Report ;A2without Search Report)
Search Report: 990210 A3 Separate publication of the European or International search report
Examination: 990922 A2 Date of request for examination: 19990723
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	9712W1	593
SPEC A	(English)	9712W1	59351
Total word count - document A			59944
Total word count - document B			0
Total word count - documents A + B			59944

INTERNATIONAL PATENT CLASS: **G06F-013/40**

...SPECIFICATION active). The bus hang recovery state machine 456 ensures that the secondary PCI bus 32 is brought back to the idle state to allow the **software** to isolate the **faulty** slot. When the hang condition is detected, the SIO 50 powers down the secondary bus slots, which would automatically place the bus 32 into the...

...chip 48 was the master) when the bus hang occurred, then the bridge chip 48 would remain on the bus. To take the bridge chip **off** the bus, the **recovery** state machine 456 forces a retry cycle on the PCI bus 32 by asserting the signal STOP(underscore).
A bus history capture block 458 monitors...

7/5,K/19 (Item 19 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00887636
Ordering transactions in a computer system
Transaktionssortierung in einem Rechnersystem
Ordonnancement des transactions dans un systeme d'ordinateur
PATENT ASSIGNEE:
Compaq Computer Corporation, (687792), 20555 S.H. 249, Houston Texas 77070, (US), (applicant designated states: AT;BE;CH;DE;DK;ES;FI;FR;GB;GR;IE;IT;LI;LU;MC;NL;PT;SE)
INVENTOR:
MacLaren, John M., 15318 Redbud Leaf, Cypress, Texas 77429, (US)
Goodrum, Alan L., 16522 Avenfield Lane, Tomball, Texas 77375, (US)
LEGAL REPRESENTATIVE:
Brunner, Michael John et al (28871), GILL JENNINGS & EVERY Broadgate House 7 Eldon Street, London EC2M 7LH, (GB)
PATENT (CC, No, Kind, Date): EP 811927 A2 971210 (Basic)
EP 811927 A3 990728
APPLICATION (CC, No, Date): EP 97303791 970604;
PRIORITY (CC, No, Date): US 655254 960605
DESIGNATED STATES: AT; BE; CH; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE
INTERNATIONAL PATENT CLASS: **G06F-013/40**

ABSTRACT EP 811927 A2
A computer system includes a first device on a first data bus, a second device on a second data bus, and a bridge device that delivers data transactions between the two devices. The bridge device includes an execution queue that stores only a higher priority transaction and transactions initiated before the higher priority transaction, and a controller that selects transactions from the execution queue to be completed on one of the data buses.
ABSTRACT WORD COUNT: 74

LEGAL STATUS (Type, Pub Date, Kind, Text):
 Withdrawal: 001004 A2 Date application deemed withdrawn: 20000129
 Application: 971210 A2 Published application (A1with Search Report ;A2without Search Report)
 Search Report: 990728 A3 Separate publication of the European or International search report
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	9712W1	682
SPEC A	(English)	9712W1	59509
Total word count - document A			60191
Total word count - document B			0
Total word count - documents A + B			60191

INTERNATIONAL PATENT CLASS: **G06F-013/40**

...SPECIFICATION active). The bus hang recovery state machine 456 ensures that the secondary PCI bus 32 is brought back to the idle state to allow the **software** to isolate the **faulty** slot. When the hang condition is detected, the SIO 50 powers down the secondary bus slots, which would automatically place the bus 32 into the...

...chip 48 was the master) when the bus hang occurred, then the bridge chip 48 would remain on the bus. To take the bridge chip **off** the bus, the **recovery** state machine 456 forces a retry cycle on the PCI bus 32 by asserting the signal STOP(underscore).
A bus history capture block 458 monitors...

7/5,K/25 (Item 25 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2003 European Patent Office. All rts. reserv.

00401208
Servicing interrupts in a data processing system
Unterbrechungsbedienung in einem Datenverarbeitungssystem
Prise en charge d'interruptions dans un systeme de traitement de donnees
PATENT ASSIGNEE:
International Business Machines Corporation, (200120), Old Orchard Road, Armonk, N.Y. 10504, (US), (applicant designated states: AT;BE;CH;DE;DK;ES;FR;GB;GR;IT;LI;LU;NL;SE)
INVENTOR:
Dinwiddie, John Monroe, Jr., 112 Pacer Circle, West Palm Beach, FL 33414, (US)
Grice, Lonnie Edward, 252 N.W. 44th Street, Boca Raton, FL 33431, (US)
Loffredo, John Mario, 2694 S.W. 14th Drive, Deerfield Beach, FL 33442, (US)
Sanderson, Kenneth Russell, 1132 Widgeon Road, West Palm Beach, FL 33414, (US)
LEGAL REPRESENTATIVE:
Bailey, Geoffrey Alan (27921), IBM United Kingdom Limited Intellectual Property Department Hursley Park, Winchester Hampshire SO21 2JN, (GB)
PATENT (CC, No, Kind, Date): EP 398696 A2 901122 (Basic)
EP 398696 A3 940105
EP 398696 B1 970723
APPLICATION (CC, No, Date): EP 90305309 900516;
PRIORITY (CC, No, Date): US 353117 890517
DESIGNATED STATES: AT; BE; CH; DE; DK; ES; FR; GB; GR; IT; LI; LU; NL; SE
INTERNATIONAL PATENT CLASS: **G06F-015/16** ; **G06F-013/26**
CITED PATENTS (EP A): US 4812975 A; US 4296466 A; EP 132157 A; EP 192944 A; EP 205949 A; EP 333617 A
CITED REFERENCES (EP A):
IBM TECHNICAL DISCLOSURE BULLETIN vol. 28, no. 12 , May 1986 , ARMONK, NY, USA pages 5326 - 5328 'PC FIXED DISK USAGE DURING EMULATION'
IEEE 1986 NATIONAL AEROSPACE AND ELECTRONICS CONFERENCE 19 May 1986 , DAYTON, USA pages 368 - 375 L. D. BROCK ET AL. 'ADVANCED INFORMATION PROCESSING SYSTEM: STATUS REPORT';

ABSTRACT EP 398696 A2

The functions of two virtual operating systems (e.g. S/370 VM, VSE or IX370 and S/88 OS) are merged into one physical system. Partner pairs of S/88 processors run the S/88 OS and handle the fault tolerant and single system image aspects of the system. One or more partner pairs of S/370 processors are coupled to corresponding S/88 processors directly and through the S/88 bus. Each S/370 processor is allocated form 1 to 16 megabytes of contiguous storage from the S/88 main storage. Each S/370 virtual operating system thinks its memory allocation starts at address 0, and it manages its memory through normal S/370 dynamic memory allocation and paging techniques. The S/370 is limit checked to prevent the S/370 from accessing S/88 memory space. The S/88 Operating System is the master over all system hardware and I/O devices. The S/88 processors across the S/370 address space in direct response to a S/88 application program so that the S/88 may move I/O data into the S/370 I/O buffers and process the S/370 I/O operations. The S/88 and S/370 peer processor pairs to execute their respective Operating Systems in a single system environment without significant rewriting of either operating system. Neither operating system is aware of the other operating system nor the other processor pairs.

ABSTRACT WORD COUNT: 214

LEGAL STATUS (Type, Pub Date, Kind, Text):

Lapse:	020612	B1	Date of lapse of European Patent in a contracting state (Country, date): AT 19970723, CH 19970723, LI 19970723, DK 19970723, ES 19970723, GR 19970723, SE 19971023,
Lapse:	20000126	B1	Date of lapse of European Patent in a contracting state (Country, date): AT 19970723, CH 19970723, LI 19970723, DK 19970723, GR 19970723, SE 19971023,
Application:	901122	A2	Published application (A1with Search Report ;A2without Search Report)
Examination:	910206	A2	Date of filing of request for examination: 901213
Search Report:	940105	A3	Separate publication of the European or International search report
Examination:	961113	A2	Date of despatch of first examination report: 960926
Grant:	970723	B1	Granted patent
Lapse:	980318	B1	Date of lapse of the European patent in a Contracting State: SE 971023
Lapse:	980408	B1	Date of lapse of the European patent in a Contracting State: AT 970723, DK 970723, SE 971023
Lapse:	980408	B1	Date of lapse of the European patent in a Contracting State: AT 970723, DK 970723, SE 971023
Oppn None:	980715	B1	No opposition filed
Lapse:	981021	B1	Date of lapse of the European patent in a Contracting State: AT 970723, CH 970723, LI 970723, DK 970723, SE 971023
Lapse:	981021	B1	Date of lapse of the European patent in a Contracting State: AT 970723, CH 970723, LI 970723, DK 970723, SE 971023

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF1	700
CLAIMS B	(English)	9707W4	715
CLAIMS B	(German)	9707W4	619
CLAIMS B	(French)	9707W4	829
SPEC A	(English)	EPABF1	70506
SPEC B	(English)	9707W4	70530
Total word count - document A			71213
Total word count - document B			72693

Total word count - docume[redacted] A + B 143906

INTERNATIONAL PATENT CLASS: **G06F-015/16** ...

... **G06F-013/26**

...SPECIFICATION that component is automatically removed from service. Processing continues on the duplexed partner while the failed component is checked by internal diagnostics. The error-detection **functions** will automatically run diagnostics on a failing component removed from service while processing continues on its duplexed partner. If the diagnostics determine that certain components...

...with selected synchronism, in the absence of any detected fault. Upon detection of an error-manifesting fault in any unit, that unit is isolated and **placed** off-line so that it cannot transfer information to other units of the module. The partner of the off-line unit continues operating, normally with...bulk supply and in turn develops the operating voltages which that unit requires. This power stage in addition monitors the supply voltages. Upon detecting a **failing** supply voltage, the power stage produces a signal that clamps to ground potential all output lines from that unit to the bus structure. This action precludes a power **failure** at any unit from causing the transmission of faulty information to the bus structure.

Some units of the processor module execute each information transfer with an operating cycle that includes an **error** -detecting timing phase prior to the actual information transfer. A unit which provides this operation, e.g. a control unit for a peripheral device, thus tests for a **fault** condition prior to effecting an information transfer. The unit inhibits the information transfer in the event a **fault** is detected. The module, however, can continue operation - without interruption or delay - and effect the information transfer from the non-inhibited partner unit.

Other units...

...at least the central processing unit and the memory unit, for which operating time is of more importance, execute each information transfer concurrently with the **error** detection pertinent to that transfer. In the event a fault is detected, the unit immediately produces a signal which alerts other processing units to disregard...

7/5,K/26 (Item 26 from file: 348)

00401206

Fault tolerant data processing system
Fehlertolerantes Datenverarbeitungssystem
Systeme de traitement de donnees a tolerance de fautes

PATENT ASSIGNEE:
International Business Machines Corporation, (200120), Old Orchard Road, Armonk, N.Y. 10504, (US), (applicant designated states: AT;BE;CH;DE;DK;ES;FR;GB;GR;IT;LI;LU;NL;SE)
INVENTOR:
Baker, Ernest Dysart, 12032 Deer Run, Raleigh, North Carolina 27614, (US)
Dinwiddie, John Monroe, Jr., 112 Pacer Circle, West Palm Beach, FL 33414, (US)
Grice, Lonnie Edward, 252 N.W. 44th Street, Boca Raton, FL 33431, (US)
Joyce, James Maurice, 1544 N.W. 9th Street, Boca Raton, FL 33486, (US)
Loffredo, John Mario, 2694 S.W. 14th Drive, Deerfield Beach, FL 33442, (US)
Sanderson, Kenneth Russell, 1132 Widgeon Road, West Palm Beach, FL 33414, (US)
Suarez, Gustavo Armando, 21482 Woodchuck Lane, Boca Raton, FL 33428, (US)
LEGAL REPRESENTATIVE:
Bailey, Geoffrey Alan (27921), IBM United Kingdom Limited Intellectual Property Department Hursley Park, Winchester Hampshire SO21 2JN, (GB)

PATENT (CC, No, Kind, Dat: EP 398694 A2 901122 (Basic)
EP 398694 A3 940202
EP 398694 B1 980909
APPLICATION (CC, No, Date): EP 90305307 900516;
PRIORITY (CC, No, Date): US 353116 890517
DESIGNATED STATES: AT; BE; CH; DE; DK; ES; FR; GB; GR; IT; LI; LU; NL; SE
INTERNATIONAL PATENT CLASS: **G06F-011/16** ; **G06F-009/44** ; **G06F-015/16**
CITED PATENTS (EP A): US 4654857 A; US 4812975 A; US 4077060 A; EP 205949 A ; EP 132157 A; EP 205943 A

ABSTRACT EP 398694 A2

The functions of two virtual operating systems (e.g. S/370 VM, VSE or IX370 and S/88 OS) are merged into one physical system. Partner pairs of S/88 processors run the S/88 OS and handle the fault tolerant and single system image aspects of the system. One or more partner pairs of S/370 processors are coupled to corresponding S/88 processors directly and through the S/88 bus. Each S/370 processor is allocated form 1 to 16 megabytes of contiguous storage from the S/88 main storage. Each S/370 virtual operating system thinks its memory allocation starts at address 0, and it manages its memory through normal S/370 dynamic memory allocation and paging techniques. The S/370 is limit checked to prevent the S/370 from accessing S/88 memory space. The S/88 Operating System is the master over all system hardware and I/O devices. The S/88 processors across the S/370 address space in direct response to a S/88 application program so that the S/88 may move I/O data into the S/370 I/O buffers and process the S/370 I/O operations. The S/88 and S/370 peer processor pairs to execute their respective Operating Systems in a single system environment without significant rewriting of either operating system. Neither operating system is aware of the other operating system nor the other processor pairs. (see image in original document)
ABSTRACT WORD COUNT: 219

LEGAL STATUS (Type, Pub Date, Kind, Text):

Lapse:	010606 B1	Date of lapse of European Patent in a contracting state (Country, date): AT 19980909, CH 19980909, LI 19980909, GR 19980909, SE 19981209,
Application:	901122 A2	Published application (A1with Search Report ;A2without Search Report)
Lapse:	020612 B1	Date of lapse of European Patent in a contracting state (Country, date): AT 19980909, CH 19980909, LI 19980909, ES 19980909, GR 19980909, SE 19981209,
Examination:	910206 A2	Date of filing of request for examination: 901213
Change:	940126 A2	Obligatory supplementary classification (change)
Search Report:	940202 A3	Separate publication of the European or International search report
Examination:	960605 A2	Date of despatch of first examination report: 960422
Grant:	980909 B1	Granted patent
Lapse:	990602 B1	Date of lapse of the European patent in a Contracting State: CH 980909, LI 980909
Lapse:	990602 B1	Date of lapse of the European patent in a Contracting State: CH 980909, LI 980909
Lapse:	990811 B1	Date of lapse of European Patent in a contracting state (Country, date): CH 19980909, LI 19980909, SE 19981209,
Lapse:	990825 B1	Date of lapse of European Patent in a contracting state (Country, date): AT 19980909, CH 19980909, LI 19980909, SE 19981209,
Oppn None:	990901 B1	No opposition filed: 19990610

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	9837	610

CLAIMS B (German) 9837 572
CLAIMS B (French) 9837 714
SPEC B (English) 9837 71492
Total word count - document A 0
Total word count - document B 73388
Total word count - documents A + B 73388

INTERNATIONAL PATENT CLASS: **G06F-011/16** ...

... **G06F-009/44** ...

... **G06F-015/16**

7/5,K/27 (Item 27 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2003 European Patent Office. All rts. reserv.

00401205
Method and apparatus for adding a data processing function to a data processing system
Verfahren und Anordnung zum Hinzufugen von einer Datenverarbeitungsfunktion zu einem Datenverarbeitungssystem
Methode et appareil pour l'addition d'un fonction de traitement des donnees a un systeme de traitement de donnees
PATENT ASSIGNEE:
International Business Machines Corporation, (200120), Old Orchard Road, Armonk, N.Y. 10504, (US), (applicant designated states: AT;BE;CH;DE;DK;ES;FR;GB;GR;IT;LI;LU;NL;SE)
INVENTOR:
Dinwiddie, John Monroe, Jr., 112 Pacer Circle, West Palm Beach, FL 33414, (US)
Grice, Lonnie Edward, 252 N.W. 44th Street, Boca Raton, FL 33431, (US)
Joyce, James Maurice, 1544 N.W. 9th Street, Boca Raton, FL 33486, (US)
Loffredo, John Mario, 2694 S.W. 14th Drive, Deerfield Beach, FL 33442, (US)
Sanderson, Kenneth Russell, 1132 Widgeon Road, West Palm Beach, FL 33414, (US)
Baker, Ernest Dysart, 12032 Deer Run, North Carolina, 27614, (US)
LEGAL REPRESENTATIVE:
Bailey, Geoffrey Alan (27921), IBM United Kingdom Limited Intellectual Property Department Hursley Park, Winchester Hampshire SO21 2JN, (GB)
PATENT (CC, No, Kind, Date): EP 398693 A2 901122 (Basic)
EP 398693 A3 940202
EP 398693 B1 980909
APPLICATION (CC, No, Date): EP 90305306 900516;
PRIORITY (CC, No, Date): US 353111 890517
DESIGNATED STATES: AT; BE; CH; DE; DK; ES; FR; GB; GR; IT; LI; LU; NL; SE
INTERNATIONAL PATENT CLASS: **G06F-015/16** ; **G06F-013/12**
CITED PATENTS (EP A): US 4812975 A; EP 205949 A; US 4354225 A; US 4315310 A ; EP 197499 A

ABSTRACT EP 398693 A2
The functions of two virtual operating systems (e.g. S/370 VM, VSE or IX370 and S/88 OS) are merged into one physical system. Partner pairs of S/88 processors run the S/88 OS and handle the fault tolerant and single system image aspects of the system. One or more partner pairs of S/370 processors are coupled to corresponding S/88 processors directly and through the S/88 bus. Each S/370 processor is allocated form 1 to 16 megabytes of contiguous storage from the S/88 main storage. Each S/370 virtual operating system thinks its memory allocation starts at address 0, and it manages its memory through normal S/370 dynamic memory allocation and paging techniques. The S/370 is limit checked to prevent the S/370 from accessing S/88 memory space. The S/88 Operating System is the master over all system hardware and I/O devices. The S/88 processors across the S/370 address space in direct response to a S/88 application program so that the S/88 may move I/O data into the S/370 I/O buffers and process the S/370 I/O operations. The S/88 and S/370 peer processor pairs

to execute their respective Operating Systems in a single system environment without significant rewriting of either operating system. Neither operating system is aware of the other operating system nor the other processor pairs. (see image in original document)
ABSTRACT WORD COUNT: 219

LEGAL STATUS (Type, Pub Date, Kind, Text):

Lapse:	010606	B1	Date of lapse of European Patent in a contracting state (Country, date): AT 19980909, CH 19980909, LI 19980909, GR 19980909, SE 19981209,
Application:	901122	A2	Published application (A1with Search Report ;A2without Search Report)
Lapse:	020612	B1	Date of lapse of European Patent in a contracting state (Country, date): AT 19980909, CH 19980909, LI 19980909, ES 19980909, GR 19980909, SE 19981209,
Examination:	910206	A2	Date of filing of request for examination: 901213
Search Report:	940202	A3	Separate publication of the European or International search report
Examination:	960724	A2	Date of despatch of first examination report: 960610
Grant:	980909	B1	Granted patent
Lapse:	990602	B1	Date of lapse of the European patent in a Contracting State: CH 980909, LI 980909
Lapse:	990602	B1	Date of lapse of the European patent in a Contracting State: CH 980909, LI 980909
Lapse:	990811	B1	Date of lapse of European Patent in a contracting state (Country, date): CH 19980909, LI 19980909, SE 19981209,
Lapse:	990825	B1	Date of lapse of European Patent in a contracting state (Country, date): AT 19980909, CH 19980909, LI 19980909, SE 19981209,
Oppn None:	990901	B1	No opposition filed: 19990610

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	9837	1109
CLAIMS B	(German)	9837	979
CLAIMS B	(French)	9837	1299
SPEC B	(English)	9837	71715
Total word count - document A			0
Total word count - document B			75102
Total word count - documents A + B			75102

INTERNATIONAL PATENT CLASS: **G06F-015/16** ...

... **G06F-013/12**

7/5,K/47 (Item 16 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

00504224 **Image available**
INFORMATION HANDLING SYSTEM WITH SUSPEND/RESUME OPERATION
SYSTEME DE GESTION D'INFORMATIONS AVEC FONCTION DE SUSPENSION/REPRISE
Patent Applicant/Assignee:
INTERNATIONAL BUSINESS MACHINES CORPORATION,
IBM UNITED KINGDOM LIMITED,
KOHNO Hiroshi,
SHIMOTOHNO Susumu,
Inventor(s):
KOHNO Hiroshi,
SHIMOTOHNO Susumu,
Patent and Priority Information (Country, Number, Date):

Patent: WO 9935576 A1 19990715
Application: WO 99GB53 19990107 (PCT/WO GB9900053)
Priority Application: JP 981253 19980107
Designated States: CZ HU IL KR PL RU US AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE
Main International Patent Class: **G06F-011/14**
Publication Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 12778

English Abstract
Provided is an information handling system and a method of controlling the same which allows the state of the system to be saved without destructing other user data on an external storage device. When a predetermined event occurs, the system saves hibernation information in a hibernation information storing area on a hard disk. A hibernation managing information storing area is provided in the outermost cylinder of the hard disk. A boot sector already exists in the outermost cylinder. Therefore, the hibernation managing information is written in the outermost cylinder after the boot sector has been saved to the hibernation information storing area. After such series of processes have been completed, the system shifts to a hibernation mode. On the other hand, when power feeding to the system is resumed, the hibernation information is read out to check whether or not the system was in the hibernation mode, etc. The hibernation information saved in the hibernation information storing area is restored to the original place and writes the master boot record which was saved to the hibernation information storing area back to the outermost cylinder.
French Abstract
Cette invention se rapporte a un systeme de gestion d'informations et a un procede de commande de ce systeme, qui permettent de sauvegarder l'etat du systeme sans detruire les autres donnees d'utilisateur stockees sur un dispositif de memorisation externe. Lorsqu'un evenement predetermine se produit, le systeme sauvegarde les informations d'hibernation dans une zone de stockage des informations d'hibernation se trouvant sur un disque dur. Une zone de stockage des informations de gestion d'hibernation est prevue dans le cylindre exterieur du disque dur. Un secteur d'initialisation se trouve deja dans ce cylindre exterieur. Par consequent, les informations de gestion d'hibernation sont inscrites dans ce cylindre exterieur, apres que le secteur d'initialisation a ete sauvegarde dans la zone de stockage des informations d'hibernation. Une fois terminee cette serie d'operations, le systeme passe en mode d'hibernation. Par ailleurs, lors du retablissement de l'alimentation du systeme, les informations d'hibernation sont extraites pour verifier si le systeme se trouve en mode d'hibernation, notamment. Les informations d'hibernation sauvegardees dans la zone de stockage des informations d'hibernation sont a nouveau stockees a leur emplacement d'origine et le fichier d'initialisation maitre, qui a ete sauvegarde dans la zone de stockage des informations d'hibernation, est inscrit dans ledit cylindre exterieur.

Main International Patent Class: **G06F-011/14**
Fulltext Availability:
Claims

Claim
... the task dares to be resumed under a different system environment.
It is desirable that the hibernation managing information storing area is in a physically **fixed** position **on** a hard disk. For example, if it is allocated in a cylinder defined in the outermost or innermost area of the hard disk, it is...includes volatile data such as the content of a memory and a VRAM, for example.
A hibernation managing information storing area is provided in a

fixed location **on** t[redacted]hard disk. For example, a cyli[redacted]r defined in the
outermost or innermost area (or a specific sector within the cylinder) of the hard disk...of the hibernation
managing information storing area fixed while the hibernation information storing area is allocated entirely independently of the physical address using a utility **program** . Nevertheless, there is no **problem** of the operation in such occasion because the data originally existing in the hibernation managing information storing area has been saved.
None of user data...50 which is
connected to an SMI pin of the CPU 11. The trap logic mainly has 2 functions. one of the functions is to **assert** the SMI signal line 50 in response to **assertion** of the control signal line 60 for generating an SMI
interruption. The other function is to continuously monitor the buses 16 and the bus 18 and **assert** the SMI signal line 50 when the address (I/O address or memory address) set in an internal register is detected for generating an SMI...hibernation managing information storing area is then assured on the hard disk (step S110).
The hibernation managing information storing area is preferably in a physically **fixed** position **on** the hard disk (a specific cylinder or a specific sector in a specific cylinder). If it is allocated in a fixed address like a cylinder...

...is required at a relatively early stage
of the wake up processing (to be described later). Accordingly, the hibernation managing information is saved in a **fixed** address **on** the hard
disk for the convenience of accessing.
The PMC then saves the allocation information of data (start address of data) in the hibernation information...example, the error processing displays an error message on the display 22 to prompt
the user to do a predetermined work. The predetermined work includes **disabling** the hibernation signature and **recovery** of the original system
configuration to restart the system 100, for example. when it is prompted to recover the system configuration, the original system configuration... such as FAT (File
Allocation Table) so long as it is a "hibernation file".
The hibernation managing information storing area exists in a place physically **fixed** **on** the hard disk (outermost or innermost cylinder or a
specific sector in the cylinder). This is because a boot sector exists in the outermost cylinder...

...the hibernation managing information storing area is fixed
while the hibernation information storing area is allocated totally independently of a physical address using a utility **program** , etc. Nevertheless, there is no **problem** of operation because the data originally
existing in the hibernation managing information storing area has been saved (Fig.7).
This invention has been described in...
?

11/5,K/3 (Item 3 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01286893

SYSTEM AND METHOD IMPROVING FAULT ISOLATION AND DIAGNOSIS IN COMPUTERS
VORRICHTUNG UND VERFAHREN ZUR VERBESSERTEN FEHLERORTUNG UND DIAGNOSE IN RECHNERN
SYSTEME ET METHODE PERMETTANT D'AMELIORER L'ISOLATION ET LE DIAGNOSTIC DE DEFAILLANCES DANS DES ORDINATEURS

PATENT ASSIGNEE:
Sun Microsystems, Inc., (2616592), 4150 Network Circle, Santa Clara, California 95054, (US), (Proprietor designated states: all)
INVENTOR:
WILLIAMS, Emrys, Helford House, Tyrells End, Evershott,, Milton Keynes MK17 9DS, (GB)
CYPHER, Robert, 18460 Montpere Way, Saratoga, CA 95070, (US)
LEGAL REPRESENTATIVE:
Harris, Ian Richard et al (72231), D. Young & Co., 21 New Fetter Lane, London EC4A 1DA, (GB)

PATENT (CC, No, Kind, Date): EP 1224548 A1 020724 (Basic)
EP 1224548 B1 030521
WO 2001025924 010412
APPLICATION (CC, No, Date): EP 2000965469 000926; WO 2000US26506 000926
PRIORITY (CC, No, Date): US 413108 991006
DESIGNATED STATES (Pub A): AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; (Pub B): AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: **G06F-011/10** ; **G06F-011/00** ; **G06F-011/22**
CITED PATENTS (EP B): US 3814922 A; US 4780809 A; US 5953351 A
CITED PATENTS (WO A): US 5953351 A ; US 3814922 A ; US 4780809 A
NOTE:
No A-document published by EPO
LEGAL STATUS (Type, Pub Date, Kind, Text):

Application:	010606 A1	International application. (Art. 158(1))
Application:	010606 A1	International application entering European phase
Application:	020724 A1	Published application with search report
Examination:	020724 A1	Date of request for examination: 20020405
Change:	021002 A1	Inventor information changed: 20020809
Change:	021016 A1	Title of invention (German) changed: 20020828
Change:	021016 A1	Title of invention (English) changed: 20020828
Change:	021016 A1	Title of invention (French) changed: 20020828
Assignee:	030423 A1	Transfer of rights to new applicant: Sun Microsystems, Inc. (2616592) 4150 Network Circle Santa Clara, California 95054 US
Grant:	030521 B1	Granted patent

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	200321	711
CLAIMS B	(German)	200321	613
CLAIMS B	(French)	200321	773
SPEC B	(English)	200321	5835
Total word count - document A			0
Total word count - document B			7932
Total word count - documents A + B			7932

INTERNATIONAL PATENT CLASS: **G06F-011/10** ...

... **G06F-011/00** ...

... **G06F-011/22**

...SPECIFICATION circuit that accepts data which does not include a flag and produces a corresponding flag. In this embodiment, data consisting of a value and an **error** detection/correction **code** is input to a

detector/corrector. The detector/corrector **checks** the value against the **error** detection/correction **code** to determine whether the value is correct. ("Correct" is used herein to describe data for which the corresponding **error** detection/correction **code** indicates no **errors** .) **If** the value is **correct** , the value is output **on** a data line while a "false" signal is asserted on a flag line. If the value is incorrect but correctable, the value is corrected and...

...on the data line while a "false" signal is asserted on the flag line. If the value is incorrect and cannot be corrected using the **error** detection/correction **code** , hardware diagnosis is initiated and a "true" signal is asserted on the flag line. When ...the uncorrectable value or a predetermined value (e.g. some value describing the detected error.) The circuit may include an ECC generator that produces an **error** detection/correction **code** corresponding to the value and the flag output by the circuit.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects and advantages of the invention will become...

11/5,K/7 (Item 7 from file: 348)

DIALOG(R)File 348:EUROPEAN PATENTS

00780859

Method and apparatus for run-time memory access checking and memory leak detection of a multi-threaded program

Verfahren und Vorrichtung zur Uberwachung der Speicherzugriffe eines Vielfadenprogramms

Methode et appareil de controle d'acces a la memoire pendant l'execution d'un programme a fils multiples

PATENT ASSIGNEE:
SUN MICROSYSTEMS, INC., (1392730), 2550 Garcia Avenue, Mountain View, CA 94043, (US), (applicant designated states: DE;FR;GB;IT;NL)
INVENTOR:
Rishi, Alok, 507 Vallejo Street, El Granada, California 94018, (US)
Masamitsu, Jon A., 1873 Creek Road, Livermore, California 94550, (US)
LEGAL REPRESENTATIVE:
W.P. Thompson & Co. (101051), Coopers Building, Church Street, Liverpool L1 3AB, (GB)

PATENT (CC, No, Kind, Date): EP 729097 A1 960828 (Basic)
APPLICATION (CC, No, Date): EP 96300759 960205;
PRIORITY (CC, No, Date): US 384884 950207
DESIGNATED STATES: DE; FR; GB; IT; NL
INTERNATIONAL PATENT CLASS: **G06F-011/00**

ABSTRACT EP 729097 A1

The present invention is a system and method for a "debugger Run-Time-Checking for valid memory accesses for multi-threaded application programs" (hereinafter "RTC/MT") wherein a run-time process which includes multiple threads running either serially or concurrently, may be monitored by a debugger program and memory access errors detected and correctly attributed to the process thread encountering the error. The RTC/MT system of the present invention also provides an apparatus and method which monitors and reports memory leaks as required for multi-threaded target programs. (see image in original document)

ABSTRACT WORD COUNT: 101

LEGAL STATUS (Type, Pub Date, Kind, Text):

Application:	960828 A1	Published application (A1with Search Report ;A2without Search Report)
Examination:	970416 A1	Date of filing of request for examination: 970214
Withdrawal:	981007 A1	Date on which the European patent application was withdrawn: 980730

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPAB96	1658
SPEC A	(English)	EPAB96	9337
Total word count - document A			10995
Total word count - document B			0
Total word count - documents A + B			10995

INTERNATIONAL PATENT CLASS: **G06F-011/00**

...SPECIFICATION 2, block 100, space is allocated for the patch tables and the patch tables and values are initialized. Next, as illustrated in block 110, the **program** to be **error checked** is initially read and loaded as it exists on the disk file. Such program is normally loaded in portions (load objects) as they are accessed...

...to run the program. Once the debugger program has received a list of the load objects, it will scan the load objects, searching for instructions **that** it is going to **patch** later **on** . The only part of the load object the debugger program looks at during this instruction-by-instruction scan are the instructions themselves, i.e., the...

11/5,K/8 (Item 8 from file: 348)

00741704

Method and apparatus for a fast debugger fix and continue operation

Verfahren und Einrichtung zur schnellen Fehlerbehebung und Arbeitsfortsetzung eines Debuggers

Procede et appareil pour depanner et continuer l'operation d'un debougeur

PATENT ASSIGNEE:
SUN MICROSYSTEMS, INC., (1392730), 2550 Garcia Avenue, Mountain View, CA 94043, (US), (applicant designated states: DE;FR;GB;IT;SE)
INVENTOR:
Preisler, Thomas, 14933 Tahoe Way, Morgan Hill, California 95037, (US)
Gramlich, Wayne C., 726 Henrietta Avenue, Sunnyvale, California 94086, (US)
Pelegri-Llopart, Eduardo, 1731 Fordham Way, Mountain View, California 94040, (US)
Miller, Terrence C., 169 Oak Court, Menlo Park, California 94025, (US)
LEGAL REPRESENTATIVE:
Johnson, Terence Leslie (42961), Edward Evans & Co. Chancery House 53-64 Chancery Lane, London WC2A 1SD, (GB)

PATENT (CC, No, Kind, Date): EP 699996 A1 960306 (Basic)
APPLICATION (CC, No, Date): EP 95305992 950829;
PRIORITY (CC, No, Date): US 299720 940901
DESIGNATED STATES: DE; FR; GB; IT; SE
INTERNATIONAL PATENT CLASS: **G06F-011/00**

ABSTRACT EP 699996 A1

This Continuation-In-part describes a part of this run-time debugger operation which is designated the "Fix-and-Continue" invention. This invention permits a user to begin a debugging session wherein if an error in the code is encountered, the user can edit the corresponding source code to correct the error and then execute a "Fix and Continue" command all without leaving the debugging session. The Fix and Continue code calls the compiler to recompile the source code file with the edited text in it, receives the resulting recompiled object code file from the compiler, uses the dynamic linker to link the recompiled object code into the target application program process, patches the previous version of this same object code file to refer to the newly recompiled code, resets any required variables and registers, resets the program counter to the line of code being executed when the error was discovered. The debugger then continues in the debug session thereby saving the time it would ordinarily take to quit the debug session, relink and reload the target program and start the debug session once

again.
ABSTRACT WORD COUNT: 198

LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: 960306 A1 Published application (A1with Search Report ;A2without Search Report)
Examination: 961106 A1 Date of filing of request for examination: 960905
Examination: 980325 A1 Date of despatch of first examination report: 980211
Withdrawal: 990120 A1 Date on which the European patent application was deemed to be withdrawn: 980623
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPAB96	1782
SPEC A	(English)	EPAB96	9983
Total word count - document A			11765
Total word count - document B			0
Total word count - documents A + B			11765

INTERNATIONAL PATENT CLASS: **G06F-011/00**

...SPECIFICATION 2, block 100, space is allocated for the patch tables and the patch tables and values are initialized. Next, as illustrated in block 110, the **program** to be **error checked** is initially read and loaded as it exists on the disk file. Such program is normally loaded in portions (load objects) as they are accessed...to run the program. Once the debugger program has received a list of the load objects, it will scan the load objects, searching for instructions **that** it is going to **patch** later **on** . The only part of the load object the debugger program looks at during this instruction-by-instruction scan are the instructions themselves, i.e., the...

11/5,K/12 (Item 12 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00516061
Programmable read only memory device having a test tool for testing the error checking and correction circuit
Programmierbarer Festwertspeicher mit Prufgerat fur den Fehlerprufungs- und korrekturschaltkreis
Dispositif de memoire fixe programmable ayant un moyen de test pour circuit de detection et de correction d'erreur
PATENT ASSIGNEE:
NEC CORPORATION, (236690), 7-1, Shiba 5-chome Minato-ku, Tokyo, (JP), (applicant designated states: DE;FR;GB)
INVENTOR:
Fukushima, Kiyoshi, c/o NEC Kyusyu, Ltd., 100, Yahatamachi, Kumamoto-shi, Kumamoto, (JP)
LEGAL REPRESENTATIVE:
Glawe, Delfs, Moll & Partner (100692), Patentanwalte Postfach 26 01 62, D-80058 Munchen, (DE)
PATENT (CC, No, Kind, Date): EP 505914 A2 920930 (Basic)
EP 505914 A3 930825
EP 505914 B1 951227
APPLICATION (CC, No, Date): EP 92104706 920318;
PRIORITY (CC, No, Date): JP 9162252 910327
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: **G06F-011/10** ; **G06F-011/26**
CITED PATENTS (EP A): EP 268289 A; FR 2633767 A; FR 2627004 A
CITED REFERENCES (EP A):
IBM TECHNICAL DISCLOSURE BULLETIN vol. 33, no. 6B, November 1990, ARMONK, NY, USA pages 135 - 136 , XP000108818 'ERROR CORRECTION CIRCUIT TESTING USING AN ON-CHIP REGISTER';

ABSTRACT EP 505914 A2

A programmable read only memory device comprises a first memory unit (11) for data codes, a second memory unit (12) for parity codes, and an error checking and correction unit (13) for retrieving an original data code from the data code and the associated parity code, wherein a test data code and an improper or proper parity code is supplied from a test data register array (19) to the error checking and correction unit so that not only the data checking circuit (13a) but also the data correction circuit (13b) are examined to see if or not any trouble takes place therein. (see image in original document)

ABSTRACT WORD COUNT: 109

LEGAL STATUS (Type, Pub Date, Kind, Text):

Application:	920930	A2	Published application (Alwith Search Report ;A2without Search Report)
Examination:	920930	A2	Date of filing of request for examination: 920318
Change:	930804	A2	Obligatory supplementary classification (change)
Search Report:	930825	A3	Separate publication of the European or International search report
Change:	940831	A2	Representative (change)
Examination:	950405	A2	Date of despatch of first examination report: 950217
Grant:	951227	B1	Granted patent
Oppn None:	961218	B1	No opposition filed

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF1	969
CLAIMS B	(English)	EPAB96	969
CLAIMS B	(German)	EPAB96	758
CLAIMS B	(French)	EPAB96	1137
SPEC A	(English)	EPABF1	3510
SPEC B	(English)	EPAB96	3600
Total word count - document A			4479
Total word count - document B			6464
Total word count - documents A + B			10943

INTERNATIONAL PATENT CLASS: **G06F-011/10** ...

... **G06F-011/26**

...SPECIFICATION not any trouble takes place in the error checking and correction circuit 3. If the read-out data code is matched with the original data **code** , the examiner judges **that** the **error checking** and **correction** circuit 3 is excellent. **If** , **on** the other hand, the data **checking** circuit 3a has a **trouble** , the data **code** is mistakenly judged to have an **error** bit, and the data correction circuit 3b corrects the **error** bit. Therefore, the data **code** read out through the 16-bit data bus system is different from the original data **code** , and the examiner diagnose the **error checking** and correction circuit 3 as defect.

However, a problem is encountered in the prior art programmable read only memory device in that any trouble in...

...SPECIFICATION out from the programmable read only memory device 1, the 32-bit data code is directly transferred to the data selector 4.

In order to **test** the **error checking** and correction circuit 3, a parity **code** is generated on the basis of a data code, and the data code and the associated parity code are written into respective addresses of the...

...the parity code are transferred to the data checking circuit 3a, and the data checking circuit 3a checks the data code and the associated parity **code** to see if or not an **error** bit is incorporated in the data **code** . Since a parity **code** is automatically generated from a data code by a parity circuit (not shown), the parity code is proper to the data code at

all times...

...3a decides the data code to be correct in so far as the programmable read only memory units 1 and 2 do not have any **trouble** . When the data **code** is correct, the data **code** is directly transferred to the data selector 4 without any correction, and is, thereafter, read out through the 16-bit data bus system 5 to the outside thereof. The data code is checked by an examiner to see if or not any **trouble** takes place in the **error checking** and correction circuit 3. If the read-out data **code** is matched with the original data **code** , the examiner judges **that** the **error checking** and **correction** circuit 3 is excellent. **If** , **on** the other hand, the data **checking** circuit 3a has a **trouble** , the data **code** is mistakenly judged to have an **error** bit, and the data correction circuit 3b corrects the **error** bit. Therefore, the data **code** read out through the 16-bit data bus system is different from the original data **code** , and the examiner diagnose the **error checking** and correction circuit 3 as defect.

However, a problem is encountered in the prior art programmable read only memory device in that any trouble in...

11/5,K/20 (Item 20 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00249490
Fail safe architecture for a computer system
Ausfallsichere Architektur fur ein Rechnersystem
Architecture sure contre les defaillances pour un systeme de calculateur
PATENT ASSIGNEE:
Mirowski, Mieczyslaw, (853140), 2405 Velvet Valley Way, Owings Mills Maryland 21117, (US), (Proprietor designated states: all)
INVENTOR:
Guziak, Robert A., 144 Kelvington, Pennsylvania 15146, (US)
Prem, Edward K., 21 Anderson Road, Glenshaw Pennsylvania 15116, (US)
LEGAL REPRESENTATIVE:
Gutmann, Ernest et al (15992), Ernest Gutmann - Yves Plasseraud S.A. 3, rue Chauveau-Lagarde, 75008 Paris, (FR)
PATENT (CC, No, Kind, Date): EP 240428 A2 871007 (Basic)
EP 240428 A3 890927
EP 240428 B1 930929
EP 240428 B2 000308
APPLICATION (CC, No, Date): EP 87400709 870331;
PRIORITY (CC, No, Date): US 846159 860331
DESIGNATED STATES: FR
INTERNATIONAL PATENT CLASS: **G06F-011/00** ; **G06F-011/26**
CITED PATENTS (EP A): EP 18736 A; US 3908099 A; WO 8502042 A; GB 2046964 A
CITED PATENTS (EP B): EP 18736 A; EP 88364 B; WO 85/02042 A; GB 2046964 A; US 3908099 A
CITED REFERENCES (EP B):
RTP23(1981)8, pages 268-275;

ABSTRACT EP 240428 A2
Fail safe architecture for a computer system.
The fail safe architecture for a computer system includes a read only memory (ROM) self-check module, a random access memory (RAM) self-check module and operation code instructions (op code) self-check module which are actuated periodically by a non-maskable interrupt (NMI) to a microprocessor. The microprocessor then suspends the current applications routine being executed. If the self-check module detects a failure, the microprocessor enters a fail safe trap routine which initially resynchronizes the operation of the microprocessor and then delays the generation of a critical timing pulse (fail safe trigger) with a series of 'jump to yourself' steps. The fail safe trigger signal activates a device which sends a fail safe square wave to a narrow bandwidth, digital, band-pass filter. If the fail safe square wave signal is not supplied to the filter during a prescribed period of time, a set of transistor switches, interposed between the computer system power supply

and the voltage regulator for the computer system, is not actuated and power is cut off to the computer system. Otherwise, if the fail safe signal is received within the prescribed window of time, switches are actuated to couple the power supply to the computer system.
ABSTRACT WORD COUNT: 205

LEGAL STATUS (Type, Pub Date, Kind, Text):

Amended:	20000308	B2 Amended patent
Application:	871007	A2 Published application (A1with Search Report ;A2without Search Report)
Amended:	20000308	B2 Date of patent maintained as amended: 20000308
Search Report:	890927	A3 Separate publication of the European or International search report
Examination:	900516	A2 Date of filing of request for examination: 900317
Examination:	920115	A2 Date of despatch of first examination report: 911202
Grant:	930929	B1 Granted patent
Oppn:	940817	B1 Opposition 01/940621 Joh. Vaillant GmbH u. Co; Berghauser Str. 40 Postfach 10 10 20; D-42850 Remscheid; (DE) (Representative:)Heim, Johann-Ludwig, Dipl.-Ing.; c/o Johann Vaillant GmbH u. Co. Postfach 10 10 20 Berghauser Strasse 40; D-42850 Remscheid; (DE)

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	200010	2013
CLAIMS B	(German)	200010	1796
CLAIMS B	(French)	200010	2211
SPEC B	(English)	200010	9762
Total word count - document A			0
Total word count - document B			15782
Total word count - documents A + B			15782

INTERNATIONAL PATENT CLASS: **G06F-011/00** ...

... **G06F-011/26**

...SPECIFICATION confirmed. The self-check module routines for the ROM, RAM and op code are discussed later with respect to Figures 2E, F and G.
Assuming **no fault** is found by the **self - check** module in step 140, the routine senses **or** monitors the **fail** safe power supply circuit sense lines. These sense lines are **described** later with respect to Figures **4A** and 4B. If the state of the sense lines is incorrect (step 142), the failure count counter is incremented (step 144) and a jump is made to fail safe trap **routine** (step 146). **If** the **correct** signal is sensed **on** the sense **lines** , step 148 changes the self- **check** pointer register to point to the **next** self- **check** module routine. In **step** 150, **the** self-check test **value** is loaded in the test register; that value is unique to the successfully executed self-check module.
Step 152 determines whether the self-check module is being executed in an initialization mode (per step 58 of Figure 2A, the **fail** safe executive routine) **and** , if so, the proper delay or the resetting of timers T2 and T1 is recognized in step 154. The self-check executive routine then jumps...

11/5,K/22 (Item 22 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2003 European Patent Office. All rts. reserv.

00242439
Job interrupt at predetermined boundary for enhanced recovery.
Jobunterbrechung an vorausbestimmter Bereichsgrenze zur verbesserten Wiederherstellung.

Interruption de tache a des frontieres predeterminees pour amelioration de reprise.

PATENT ASSIGNEE:
International Business Machines Corporation, (200120), Old Orchard Road, Armonk, N.Y. 10504, (US), (applicant designated states: DE;FR;GB;IT)
INVENTOR:
Collins, Robert William, 2404, 4th Avenue, N.W., Rochester, MN 55901, (US)
Davidson, Wiliam Simpson, Box 293, Route 2, Oronoco, MN 55960, (US)
Dickes, Steven Michael, 3528 21st Avenue N.W., Rochester, MN 55901, (US)
Effle, James Scott, 4024 2nd Street, N.W., Rochester, MN 55901, (US)
Larson, Carle John, 2807 4th Avenue, N.W., Rochester, MN 55901, (US)
Weinschenk, Russell John, 417 19 th Avenue, S.W., Rochester, MN 55902, (US)
LEGAL REPRESENTATIVE:
Vekemans, Andre (18921), Compagnie IBM France Departement de Propriete Intellectuelle, F-06610 La Gaude, (FR)

PATENT (CC, No, Kind, Date): EP 249061 A2 871216 (Basic)
EP 249061 A3 891129
EP 249061 B1 931020
APPLICATION (CC, No, Date): EP 87107232 870519;
PRIORITY (CC, No, Date): US 873909 860613
DESIGNATED STATES: DE; FR; GB; IT
INTERNATIONAL PATENT CLASS: **G06F-011/14**
CITED PATENTS (EP A): GB 1178653 A; FR 2435757 A

ABSTRACT EP 249061 A2

A recovery mechanism restarts jobs following correction of a system failure and automatically marks the jobs for interruption at a logical boundary. The logical boundary is above logical file updating functions such that logical files are in a known state when jobs reach the boundary. When a system failure is detected which has not yet resulted in lost data, an image of working memory, including hardware status is saved on nonvolatile storage. After the failure has been resolved, the system is initially loaded with operating programs (IPL) and working memory is reloaded from the nonvolatile storage. All jobs which were reloaded are marked for interrupt at a machine instruction boundary, and processing is started. After all jobs have reached the boundary, or a predetermined time has elapsed, processing is stopped and the system is re-IPLed. There are few system index recoveries to be performed, since most jobs reached a point where logical files were synchronized with corresponding data.

ABSTRACT WORD COUNT: 162

LEGAL STATUS (Type, Pub Date, Kind, Text):

Application:	871216	A2	Published application (Alwith Search Report ;A2without Search Report)
Examination:	880629	A2	Date of filing of request for examination: 880426
Search Report:	891129	A3	Separate publication of the European or International search report
Examination:	920205	A2	Date of despatch of first examination report: 911223
Grant:	931020	B1	Granted patent
Lapse:	940810	B1	Date of lapse of the European patent in a Contracting State: DE 931020
Oppn None:	941012	B1	No opposition filed
Lapse:	950118	B1	Date of lapse of the European patent in a Contracting State: DE 931020, FR 940311
Lapse:	950510	B1	Date of lapse of the European patent in a Contracting State: DE 931020, FR 940311, GB 940519
Lapse:	991020	B1	Date of lapse of European Patent in a contracting state (Country, date): DE 19931020, FR 19940311, GB 19940519, IT 19931020,

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:
Available Text Language Update Word Count

CLAIMS B (English) EPBBF1 394
CLAIMS B (German) EPBBF1 553
CLAIMS B (French) EPBBF1 579
SPEC B (English) EPBBF1 4953
Total word count - document A 0
Total word count - document B 6479
Total word count - documents A + B 6479

INTERNATIONAL PATENT CLASS: **G06F-011/14**

...SPECIFICATION have been interrupted.
Fig. 6 is a detailed flow diagram describing the flow in VMC machine check handler 226 (Fig. 2) as it relates to **the** present invention. Upon setting of the **machine check** flag **or** counter at 612 in the HMC machine check handler, the VMC machine check handler is started at 614. **If** the **machine check code** passed **by** the HMC machine **check** handler is an 823 at decision block 616, control is passed back to the HMC machine check handler at 618 because the VMC recognizes that...

11/5,K/23 (Item 1 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

01013792
ERROR CORRECTING MEMORY AND METHOD OF OPERATING SAME
MEMOIRE DE CORRECTION D'ERREURS ET PROCEDE DE MISE EN OEUVRE DE LA MEMOIRE
Patent Applicant/Assignee:
MONOLITHIC SYSTEM TECHNOLOGY INC, 1020 Stewart Drive, Sunnyvale, CA 94085-3914, US, US (Residence), US (Nationality)
FREEMAN Jacqueline Carol, W.P. Thompson & Co., Celcon House, 289-293 High Holborn, London WC1V 7HU, GB, GB (Residence), GB (Nationality), (Designated only for: MG)
Inventor(s):
LEUNG Wingyu, 10450 Orange Avenue, Cupertino, CA 95014, US,
HSU Fu-Chieh, 21775 Congress Hall Lane, Saratonga, CA 95070-9714, US,
Legal Representative:
FREEMAN Jacqueline Carol (agent), W.P. Thompson & Co., Celcon House, 289-293 High Holborn, London WC1V 7HU, GB,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200342826 A2 20030522 (WO 0342826)
Application: WO 2002GB5123 20021113 (PCT/WO GB0205123)
Priority Application: US 20013602 20011114
Designated States: JP MG
(EP) AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LU MC NL PT SE SK TR
Main International Patent Class: **G06F-011/10**
Publication Language: English
Filing Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 10523

English Abstract
A memory device that uses error correction code (ECC) circuitry to improve the reliability of the memory device in view of single-bit errors caused by hard failure or soft error. A write buffer is used to post write data, so that ECC generation and memory write array operation can be carried out in parallel. As a result there is no penalty in write latency or memory cycle time due to ECC generation. A write-back buffer is used to post corrected ECC words during read operations, so that write-back of corrected ECC words does not need to take place during the same cycle that data is read. Instead, write-back operations are performed during idle cycles when no external memory access is requested, such that the write back operation does not impose a penalty on memory cycle time or affect memory access latency.
French Abstract
L'invention concerne un dispositif a memoire, qui utilise un circuit a

code de correction d'erreurs (ECC) pour ameliorer la fiabilite du dispositif a memoire, a la suite d'erreurs portant sur un seul bit, causees par une panne machine ou une erreur intermittente. Un tampon d'ecriture est utilise pour inscrire des donnees d'ecriture, si bien que la generation du code de correction d'erreurs et l'operation d'ecriture en memoire orientee tableau sont menees en parallele. Il en resulte une absence de penalite en termes de temps d'attente d'ecriture ou de temps du cycle memoire due a la generation du code de correction d'erreurs. Un tampon de reecriture est utilise pour inscrire des mots EEC corrigees pendant les operations de lecture, si bien que la reecriture des mots EEC corrigees n'est pas necessaire dans le meme cycle que la lecture des donnees. En revanche, des operations de reecriture sont effectuees pendant les cycles d'inactivite quand un acces exterieur a la memoire n'est pas demande, de sorte que l'operation de reecriture n'impose pas de penalite au temps du cycle memoire ni n'a d'incidence sur le temps d'attente d'acces a la memoire.

Legal Status (Type, Date, Text)
Publication 20030522 A2 Without international search report and to be republished upon receipt of that report.
Main International Patent Class: **G06F-011/10**
Fulltext Availability:
Detailed Description

Detailed Description
... sequence, or
during a write-ba ck operation. As a result, the error correction scheme does,not increase the read@latency of the memory.- Similarly, **error** correction **check** bits are only generated during refresh **operations** of the memory.

As a result, the generation of **error** correction check bits does not increase the write latency of the memory.

However, this error correction scheme cannot correct data errors occurring in the first...

11/5,K/31 (Item 9 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

00816744 **Image available**
COOPERATIVE ERROR HANDLING SYSTEM
SYSTEME COLLECTIF DE TRAITEMENT D'ERREURS
Patent Applicant/Assignee:
INTEL CORPORATION, 2625 Walsh Avenue, Santa Clara, CA 95051, US, US (Residence), US (Nationality)
Inventor(s):
MARISETTY Suresh, 1662 Lederer Circle, San Jose, CA 95131, US,
AYYAR Mani, 18816 Tuggle Avenue, Cupertino, CA 95014, US,
QUACH Nhon, 6522 Pfeiffer Ranch Road, San Jose, CA 95120, US,
LINT Bernard, 445 Poppy Place, Mountain View, CA 94043, US,
Legal Representative:
MCCRACKIN Ann (agent), Schwegman, Lundberg, Woessner & Kluth, P.O. Box 2938, Minneapolis, MN 55402, US,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200150262 A1 20010712 (WO 0150262)
Application: WO 2000US35660 20001229 (PCT/WO US0035660)
Priority Application: US 99475417 19991230
Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR
(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM

Main International Patent Class: **G06F-011/00**
International Patent Class: **G06F-011/14**
Publication Language: English
Filing Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 7496

English Abstract
Systems and methods for error handling are disclosed. The systems and methods may be utilized for single or multiple processor computer systems to handle errors in a coordinated manner between hardware and any firmware or software layers. A computer system includes a non volatile memory and at least one processor. A firmware error handling routine is stored on the non volatile memory. The firmware error handling routine is for handling errors. Each of the at least one processors detects errors. Each processor executes the firmware error handling routing on detecting error. The executed firmware error handling routing handles the error. The executed firmware error handling routine also logs error information to a log. The systems and methods provide for coordinated error handling that enhance error recovery, provide error containment and maintain system availability.

French Abstract
L'invention concerne des systemes et des procedes de traitement d'erreurs pouvant etre utilises avec des systemes informatiques comprenant un ou plusieurs processeurs, pour traiter de maniere coordonnee les erreurs entre le materiel et toute couche micrologicielle ou logicielle. L'invention concerne un systeme informatique comprenant une memoire non volatile et au moins un processeur. Un sous-programme micrologiciel de traitement d'erreurs est mis en memoire dans la memoire non volatile, il permet de traiter les erreurs. Chacun des processeurs permet de detecter les erreurs. Chacun de ces processeurs utilise le sous-programme micrologiciel de traitement d'erreurs pour detecter les erreurs. L'execution du sous-programme susmentionne permet de traiter les erreurs. Ce sous-programme enregistre egalement les informations d'erreurs dans un journal. Les systemes et les procedes decrits dans la presente invention permettent un traitement coordonne des erreurs, ce qui ameliore la reprise sur incident, le confinement des erreurs et la capacite de mise a jour du systeme.

Legal Status (Type, Date, Text)
Publication 20010712 A1 With international search report.
Publication 20010712 A1 Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.
Examination 20011122 Request for preliminary examination prior to end of 19th month from priority date

Main International Patent Class: **G06F-011/00**
International Patent Class: **G06F-011/14**
Fulltext Availability:
Claims

Claim
... to reboot on failure to correct the error.
I S. A method for cooperative error handling in a computer system comprising:
1 0 detecting an **error** by a detecting processor;
executing **error** handling **code** of a first layer of **software** , by the detecting processor, to perforin the following:
saving state infori-nation;
attempting to correct the **error** ;
on **failure** to correct the **error** , executing **error** handling **code** of a
second layer of **software** by the detecting processor to perforin the following:

determining severity of error by analyzing state
information and the error received from the first layer;
2 0 saving additional state inforination; and
halting the computer system **if** necessary; and
on **failure** to **correct** the **error** by the second layer of **software** ,
executing **error** handling **code** of an operating system by the detecting
processor to perforrn the following:
2 5 **checking** state information and the **error** to see if
processing can continue;
halting the computer system if processing unless
processing can continue; and
attempting to correct the error.
3 0

19...

11/5,K/35 (Item 13 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

00792423 **Image available**
MECHANISM TO IMPROVE FAULT ISOLATION AND DIAGNOSIS IN COMPUTERS
MECANISME PERMETTANT D'AMELIORER L'ISOLATION ET LE DIAGNOSTIC DE DEFAILLANCES DANS DES ORINATEURS
Patent Applicant/Assignee:
SUN MICROSYSTEMS INC, 901 San Antonio Road, Palo Alto, CA 94303, US, US (Residence), US (Nationality)
Inventor(s):
WILLIAMS Emrys, 1063 Morse Avenue, #3-205, Sunnyvale, CA 94089, US,
CYPHER Robert, 18460 Montpere Way, Saratoga, CA 95070, US,
Legal Representative:
KIVLIN B Noel (agent), Conley, Rose & Tayon, P.C., P.O. Box 398, Austin, TX 78767-0398, US,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200125924 A1 20010412 (WO 0125924)
Application: WO 2000US26506 20000926 (PCT/WO US0026506)
Priority Application: US 99413108 19991006
Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE
(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM
Main International Patent Class: **G06F-011/10**
International Patent Class: **G06F-011/00 ; G06F-011/22**
Publication Language: English
Filing Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 6977

English Abstract
A system and method for improving the isolation and diagnosis of hardware faults in a computing system wherein means are provided for indicating whether unusable data has previously triggered diagnosis of the hardware fault that caused the data to be unusable. If diagnosis has not been performed, the flag is not set. If diagnosis has already been performed, the flag is set. One embodiment comprises an interface which is used to convey data from one subsystem to another. When the interface receives data from the first subsystem, the data is examined to determine whether it contains an uncorrectable error (including missing data). If the data contains an uncorrectable error, the interface examines the flag corresponding to the data to determine whether hardware fault diagnosis has already been initiated. If diagnosis has already been initiated, the

data is passed to the second subsystem without initiating further diagnosis. If diagnosis has not been initiated, the interface initiates diagnosis and sets the flag to indicate that diagnosis has already been initiated. The data and corresponding flag are then passed to the second subsystem. If the data contains an uncorrectable error, data error handling procedures will be performed by the subsystem that requested the data, regardless of the value of the corresponding flag.

French Abstract

La presente invention concerne un systeme et un procede permettant d'ameliorer l'isolation et le diagnostic de defaillances materielles dans un systeme informatique qui comprend des moyens permettant d'indiquer si des donnees inutilisables ont precedemment declenche un diagnostic de defaillance materielle rendant les donnees inutilisables. Si le diagnostic n'a pas ete effectue, le drapeau n'apparait pas. Si le diagnostic a deja ete effectue, le drapeau apparait. Un mode de realisation comprend une interface destinee a acheminer des donnees d'un sous-systeme a l'autre. Lorsque l'interface recoit des donnees provenant du premier sous-systeme, lesdites donnees sont examinees afin de determiner si elles contiennent une erreur qui ne peut pas etre corrigee (y compris les donnees manquantes). Si les donnees comprennent une erreur qui ne peut pas etre corrigee, l'interface examine le drapeau correspondant aux donnees afin de determiner si le diagnostic de defaillance materielle a deja ete mis en action. Si le diagnostic a deja mis en action, les donnees sont passees vers le second sous-systeme sans entreprendre de diagnostic supplementaire. Si le diagnostic n'a pas ete mis en action, l'interface met en action le diagnostic et fait apparaitre le drapeau afin d'indiquer que le diagnostic a deja ete effectue. Les donnees et le drapeau correspondant sont ensuite achemines vers le second sous-systeme. Si les donnees contiennent une erreur qui ne peut pas etre corrigee, les procedures de traitement de donnees comportant une erreur seront effectuees par le sous-systeme demandeur de donnees, sans tenir compte de la valeur du drapeau correspondant.

Legal Status (Type, Date, Text)
Publication 20010412 A1 With international search report.
Examination 20010802 Request for preliminary examination prior to end of 19th month from priority date

Main International Patent Class: **G06F-011/10**
International Patent Class: **G06F-011/00** ...

... **G06F-011/22**
Fulltext Availability:
Detailed Description

Detailed Description

... circuit that accepts data which does not include a flag and produces a corresponding flag. In this embodiment, data consisting of a value and an **error** detection/correction **code** is input to a detector/corrector. The detector/corrector **checks** the value against the **error** detection/correction **code** to determine whether the value is correct. ("Correct" is used herein to describe data for which the corresponding **error** detection/correction **code** indicates no **errors** .) **If** the value is **correct** , the value is output **on** a data line while a "false" signal is asserted on a flag line. If the value is incorrect but correctable, the value is corrected and...

...on the data line while a "false" signal is asserted on the flag line. If the value is incorrect and cannot be corrected using the **error** detection/correction **code** , hardware diagnosis is initiated and a "true" signal
3
is asserted on the flag line. When the "true" signal is asserted, the data line may...

...the uncorrectable value or a predetermined value (e.g. some value describing the detected error.) The circuit may include an ECC generator

```
File   8:Ei Compendex(R) 1970-2003/Jun W1
         (c) 2003 Elsevier Eng.  Info. Inc.
File  35:Dissertation Abs Online 1861-2003/May
         (c) 2003 ProQuest Info&Learning
File 202:Info. Sci. & Tech. Abs. 1966-2003/May 14
         (c) Information Today, Inc
File  65:Inside Conferences 1993-2003/Jun W2
         (c) 2003 BLDSC all rts. reserv.
File   2:INSPEC 1969-2003/Jun W1
         (c) 2003 Institution of Electrical Engineers
File 233:Internet & Personal Comp. Abs. 1981-2003/May
         (c) 2003 Info. Today Inc.
File  94:JICST-EPlus 1985-2003/Jun W2
         (c)2003 Japan Science and Tech Corp(JST)
File 603:Newspaper Abstracts 1984-1988
         (c)2001 ProQuest Info&Learning
File 483:Newspaper Abs Daily 1986-2003/Jun 10
         (c) 2003 ProQuest Info&Learning
File   6:NTIS 1964-2003/Jun W2
         (c) 2003 NTIS, Intl Cpyrght All Rights Res
File 144:Pascal 1973-2003/May W4
         (c) 2003 INIST/CNRS
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 1998 Inst for Sci Info
File  34:SciSearch(R) Cited Ref Sci 1990-2003/Jun W1
         (c) 2003 Inst for Sci Info
File  99:Wilson Appl. Sci & Tech Abs 1983-2003/Apr
         (c) 2003 The HW Wilson Co.
File 583:Gale Group Globalbase(TM) 1986-2002/Dec 13
         (c) 2002 The Gale Group
File 266:FEDRIP 2003/Apr
         Comp & dist by NTIS, Intl Copyright All Rights Res
File  95:TEME-Technology & Management 1989-2003/May W4
         (c) 2003 FIZ TECHNIK
File 438:Library Lit. & Info. Science 1984-2003/Apr
         (c) 2003 The HW Wilson Co
File 239:Mathsci 1940-2003/Jul
         (c) 2003 American Mathematical Society

Set     Items   Description
S1      42878   ASSERTION? ? OR ASSERT
S2     223906   (ERROR? ? OR FAIL? OR FAULT? ? OR PROBLEM? ? OR TROUBLE OR
              FLAW OR IRREGULAR? OR GLITCH?? OR FALSE OR ODD???? OR STRANGE-
              ???? OR ANOMAL? OR ABNORMAL?)(5N)(TEST??? OR CHECK???)
S3     338401   (ERROR? ? OR FAIL? OR FAULT? ? OR PROBLEM? ? OR TROUBLE OR
              FLAW OR IRREGULAR? OR GLITCH?? OR FALSE OR ODD???? OR STRANGE-
              ???? OR ANOMAL? OR ABNORMAL?)(5N)(CONDITION? ? OR STATE OR ST-
              ATES OR SITUATION OR STATUS)
S4     470262   (RECOVER? OR CORRECT? OR FIX??? OR MEND??? OR REMED??? OR -
              RECTIF? OR REPAIR? OR PATCH? OR RESTOR? OR RESOLV? OR SOLV?)(-
              5N)(ENABL? OR ON OR DISABL? OR OFF)
S5       7361   ((INSERT? OR PUT???? OR PLAC??? OR PLACEMENT OR ADD??? OR -
              APPEND?)(5N)(CODE? ? OR INSTRUCTION? ? OR FUNCTION? ? OR COMM-
              AND? ? OR ROUTINE? ? OR PROCEDURE? ?))(5W)(PROGRAM? ? OR CODE
              OR APPLICATION? ? OR SOFTWARE)
S6     499572   (ERROR? ? OR FAIL? OR FAULT? ? OR PROBLEM? ? OR TROUBLE OR
              FLAW OR IRREGULAR? OR GLITCH?? OR FALSE OR ODD???? OR STRANGE-
              ???? OR ANOMAL? OR ABNORMAL?)(10N)(PROGRAM? ? OR CODE OR APPL-
              ICATION? ? OR SOFTWARE OR INSTRUCTIONS OR OPERATIONS)
S7         59   S1 AND S4:S5 AND S6
S8         44   RD (unique items)
S9         38   S8 NOT PY=2001:2003
S10       933   S2 AND S4 AND S6
S11     27362   (WHEN OR IF OR SHOULD OR WHILE OR THAT)(5W)S4
S12       107   S10 AND S11
S13        75   RD (unique items)
S14        70   S13 NOT (S9 OR PY=2001:2003)
S15       107   S2 AND S5 AND S6
```

S16	81	RD (unique items)
S17	67	S16 NOT (S9 OR S14 OR PY=2001:2003)
S18	62	S3 AND S11 AND S6
S19	46	RD (unique items)
S20	36	S19 NOT (S9 OR S14 OR S17 OR PY=2001:2003)
S21	7	S1:S3 AND S4 AND S5
S22	5	RD (unique items)

9/5/1 (Item 1 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

05241954 E.I. No: EIP99034593481
Title: Redundant exception check elimination by assertions
Author: Sato, Norio
Corporate Source: NTT Optical Network Systems Lab, Musashino-shi, Jpn
Source: IEICE Transactions on Communications v E81-B n 10 Oct 1998. p 1881-1893
Publication Year: 1998
CODEN: ITRCEC ISSN: 0916-8516
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 9904W4
Abstract: Exception handling is not only useful for increasing program readability, but also provides an effective means to check and locate **errors** , so it increases productivity in large-scale **program** development. Some typical and frequent **program errors** , such as out-of-range indexing, null dereferencing, and narrowing violations, cause exceptions that are otherwise unlikely to be caught. Moreover, the absence of a catcher for exceptions thrown by API procedures also causes uncaught exceptions. This paper discusses how the exception handling mechanism should be supported by the compiler together with the operating system and debugging facilities. This mechanism is implemented in the compiler by **inserting** inline check **code** and accompanying propagation **code** . One drawback to this approach is the runtime overhead imposed by the inline check code, which should therefore be optimized. However, there has been little discussion of appropriate optimization techniques and efficiency in the literature. Therefore, a new solution is proposed that formulates the optimization **problem** as a common **assertion** elimination (CAE). **Assertions** consist of check **code** and useful branch conditions. The latter are effective to remove redundant check code. The redundancy can be checked and removed precisely with a forward iterative data flow analysis. Even in performance-sensitive applications such as telecommunications software, figures obtained by a CHILL optimizing compiler indicate that CAE optimizes the code well enough to be competitive with check suppressed code. (Author abstract) 27 Refs.
Descriptors: **Software** engineering; **Error** detection; **Error** correction; **Program** compilers; **Program** debugging; Computer operating systems; Codes (symbols); Optimization; Iterative methods; Data flow analysis
Identifiers: Exception handling support; Inline check code; Common **assertion** elimination
Classification Codes:
723.1.1 (Computer Programming Languages)
723.1 (Computer Programming); 721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory); 723.2 (Data Processing); 921.5 (Optimization Techniques); 921.6 (Numerical Methods)
723 (Computer Software); 721 (Computer Circuits & Logic Elements); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

9/5/3 (Item 3 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

00738880 E.I. Monthly No: EI7808056104 E.I. Yearly No: EI78015716
Title: EXECUTABLE ASSERTIONS **-- AN AID TO RELIABLE SOFTWARE.**
Author: Saib, S. H.
Corporate Source: Gen Res Corp, Santa Barbara, Calif
Source: Rec Asilomar Conf Circuits Syst Comput 11th, Pacific Grove, Calif, Nov 7-9 1977. Publ by IEEE (77CH1315-1 C/CAS), New York, NY, 1978. Available from IEEE Comput Soc, Long Beach, Calif p 277-281
Publication Year: 1977
CODEN: RACSDI ISSN: 0736-5861
Language: ENGLISH

Journal Announcement: 7

Abstract: Two preprocessors, one for FORTRAN and one for PASCAL, have been implemented to allow " executable **assertions** " to be **added** to the source **code** . These **assertions** make it possible to carry out certain static and dynamic checks on the semantics of a **program** . They can detect **errors** in input data and prevent **error** propagation. It is possible to include remedial **instructions** to compensate for detected **errors** and provide **fault** tolerance. The **assertions** can be applied to all the data types available in the languages. Executable **assertions** can also be used in a proof of correctness. The syntax of the **assertions** , which contain first-order logical expressions is described. Examples are given of their use in a simple, but practical, example; the calculation of the time that a ballistic projectile travels. 6 refs.

Descriptors: *COMPUTER PROGRAMMING LANGUAGES; COMPUTER PROGRAMMING; COMPUTER OPERATING SYSTEMS--Program Processors

Identifiers: SOFTWARE

Classification Codes:
723 (Computer Software)
72 (COMPUTERS & DATA PROCESSING)

9/5/4 (Item 1 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01793491 ORDER NO: AADAA-I1401984

Assertion **seeding: Development of program instrumentation through iterative formal analysis**

Author: Nagulakonda, Vikram
Degree: M.S.E.E.
Year: 1999
Corporate Source/Institution: West Virginia University (0256)
Chair: John R. Callahan
Source: VOLUME 39/02 of MASTERS ABSTRACTS.
PAGE 530. 80 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984
ISBN: 0-599-99231-X

Model Checking has been successfully used for validating the requirements of concurrent systems, but validating requirements does not guarantee a sound implementation. We combine the power of model checking with source code **assertions** to verify concurrent systems. **Assertions** have been used for detecting **software faults** during debugging and for run-time checking in production releases of software. We developed the Model-based **Assertion** Generator and Extraction Tool (<italic>MAGET</italic>) to extract a PROMELA state model from an **assertion** -annotated Java program. The extracted model is then validated against properties specified in linear temporal logic (LTL) and the results of the validation process are used to add or modify **assertions** in the Java program. This approach serves multiple purposes of verifying the implementation and provides guidance in **placing assertions** in the source **code** . We illustrate our approach on a Java simulation of Shuttle Liquid Hydrogen Tanking Subsystem and discuss the results.

9/5/6 (Item 3 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01562381 ORDER NO: AAD97-17585

FORMAL METHODS FOR DESIGN, DEVELOPMENT, AND RUNTIME: RUNTIME VERIFICATION OF DISTRIBUTED REACTIVE SYSTEMS USING DR-VIA AND RTV WITH EXTENDED TTM/RTTL NOTATION (PROVABILITY, SYNCHRONIZATION, CORRECTNESS)

Author: GRASSO, CHRISTOPHER ANTHONY
Degree: PH.D.
Year: 1996
Corporate Source/Institution: UNIVERSITY OF COLORADO AT BOULDER (0051)

Director: RENJENG SU
Source: VOLUME 58/01-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 334. 201 PAGES
Descriptors: ENGINEERING, ELECTRONICS AND ELECTRICAL ; COMPUTER SCIENCE ; ENGINEERING, SYSTEM SCIENCE
Descriptor Codes: 0544; 0984; 0790

Formal specification is useful for designing provably correct models of distributed real-time software systems. Such systems are difficult to design and verify for a variety of reasons, among them: fast response times for **correct** behavior, dependence **on** unreproducible real-world events, unreported failures, intermittent communications, and mission-threatening failure modes. Some of these problems can be overcome using formal specification.

The Design and Runtime Verification, Implementation, and Analysis (DR-VIA) design methodology proposed here identifies flaws in systems under design which violate requirements. It utilizes the design model through all phases of a project, including verifying the runtime behavior of the system. Identifying flaws early reduces the cost of implementation and improves system reliability.

DR-VIA mathematically models components of a real-time system as timed transition modules (TTMs), with transitions leading from activity to activity. Event synchronization between separate modules is accomplished using shared event nomenclature for synchronous and asynchronous events. A new capability allows potential event communications failures to model failures in the actual system. The mathematical model is analyzed, not simulated, using a search engine to test safety and behavioral **assertions** .

The search engine is used during runtime on telemetry received from the operating distributed system. It efficiently verifies that the telemetry represents a safe system state. A continuous-time representation of the system components is introduced which makes efficient runtime proofs possible, a capability lacking in other real-time analysis software.

This thesis details a methodology which extends formal specification from the analytical domain to the runtime domain. It includes design analysis for correctness and introduces a means to identify synchronization failure effects. During implementation, the model is modularized into object-oriented components. The model iteratively developed during design coordinates runtime **operations** , predicting system behavior if **failures** occur.

Timed transition module extensions are introduced which allow the asynchronous, faked, and failed events, thereby modeling real-world failures. The Real-Time Verifier, an analysis tool written by the author, is discussed and demonstrated. RTV is implemented in Prolog (ECLiPSe), and enables rapid system analysis. Mathematical representations and algorithms are outlined. Sample **problems** illustrate operation of the **software** . Performance benchmarks are presented, and enhancements are discussed.

9/5/10 (Item 7 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

1070996 ORDER NO: AAD89-15447
AN EXPERIMENT IN SOFTWARE FAULT **ELIMINATION AND** FAULT **TOLERANCE**
Author: SHIMEALL, TIMOTHY JAY
Degree: PH.D.
Year: 1989
Corporate Source/Institution: UNIVERSITY OF CALIFORNIA, IRVINE (0030)
CHAIR: NANCY G. LEVESON
Source: VOLUME 50/05-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 2028. 146 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

Computer system reliability has been a concern of computer scientists and professionals since the inception of the field. Three primary approaches have been taken in developing methods to improve **software**

reliability: **fault** avoidance, **fault** elimination and **fault** tolerance. Evaluation of these approaches has for the most part been done in a vacuum; that is, there has been no comparison between different approaches. However, in any real project, all of these will be used and the real problem is to determine how to allocate scarce resources to different techniques. This study investigates the **error** detection obtained by **application** of two of these approaches, **fault** tolerance and **fault** elimination, on a set of independently developed versions of a **program** . Different **fault** detection techniques following each approach are used to provide a broad exposure of each approach on the versions. The fault detection techniques chosen were multi-version voting, programmer- **inserted** run-time **assertions** , testing, **code** reading of uncommented **code** by stepwise abstraction and static data flow analysis. Voting and run-time **assertions** are most commonly associated with **fault** tolerance. Testing, **code** reading and static data flow analysis are most commonly associated with **fault** elimination. After **application** of the techniques following each approach, the **errors** detected and the circumstances of detection were analyzed as a means of characterizing the differences between the approaches.

The results of this study provide insight on a series of research questions. The results demonstrate weaknesses in the fault tolerance approach and specifically in the multi-version voting method. In particular, the results demonstrate that voting of untested software may produce an insufficient improvement in the probability of producing a correct result to consider such use in systems where reliability is important. Voting is thus shown not to be a substitute for testing. Examination of the faults detected in this experiment show that the majority of faults were detected by only one technique. The results of this study suggest a series of questions for further research. For example, research is needed on how to broaden the classes of faults detected by each technique.

9/5/11 (Item 8 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

936399 ORDER NO: AAD86-26429
STRUCTURE DESIGN FOR FAULT **TOLERANT DISTRIBUTED** PROGRAMS **(GLOBAL** ASSERTIONS **, COMMUNICATION-CLOSED LAYERS, SAFE LAYERING, INTERMEDIATE SCHEME)**
Author: LEE, PEN-NAN
Degree: PH.D.
Year: 1986
Corporate Source/Institution: ILLINOIS INSTITUTE OF TECHNOLOGY (0091)
Source: VOLUME 47/08-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 3430. 182 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

In this thesis, we investigate a technique for specifying distributed program modules for dynamically detecting and recovering from violations of safety properties. The technique is based on performing global **assertions** based **on** a proof of **correctness** at run time. Based **on** Communication-closed Layers, we suggest the layer as the syntactic unit for recovery, and establish a generic representation and methodology called SALVIGA (SAfe Layering VIrtual Global **Assertion**). It permits hierarchical **fault** -tolerance of a wide range of properties of distributed **programs** . An intermediate buffer scheme is utilized to provide asynchronous computation, to permit **error** recovery and to form a hierarchy of distributed **software** systems. We also suggest using an N-version block--a technique that combines Recovery-block and N-version programming--for private (local) fault tolerance.

9/5/13 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC

6893700 INSPEC Abstract Number: C2001-05-6150G-041
Title: A language framework for expressing checkable properties of dynamic software
Author(s): Corbett, J.C.; Dwyer, M.B.; Hatcliff, J.; Robby
Author Affiliation: Hawaii Univ., Honolulu, HI, USA
Conference Title: SPIN Model Checking and Software Verification. 7th International SPIN Workshop. Proceedings (Lecture Notes in Computer Science Vol.1885) p.205-23
Editor(s): Havelund, K.; Penix, J.; Visser, W.
Publisher: Springer-Verlag, Berlin, Germany
Publication Date: 2000 Country of Publication: Germany x+342 pp.
ISBN: 3 540 41030 9 Material Identity Number: XX-2000-02595
Conference Title: SPIN Model Checking and Software Verification. 7th International SPIN Workshop
Conference Sponsor: Res. Inst. Adv. Comput. Sci. (RIACS)
Conference Date: 30 Aug.-1 Sept. 2000 Conference Location: Stanford, CA, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: Research **on** how to reason about **correctness** properties of software systems using model checking is advancing rapidly. Work on extracting finite state models from program source code and on abstracting those models is focused on enabling the tractable checking of program properties such as freedom from deadlock and **assertion** violations. For the most part, the **problem** of specifying more general **program** properties has not been considered. The authors report on support for specifying properties of dynamic multi-threaded Java programs that we have built into the Bandera system. Bandera extracts finite state models, in the input format of several existing model checkers, from Java code based on the property to be checked. The Bandera Specification Language (BSL) provides a language for defining general **assertions** and pre/post conditions on methods. It also supports the definition of observations that can be made on the state of program objects and the incorporation of those observations as predicates that can be instantiated in the scope of object quantifiers and used in describing common forms of state/event sequencing properties. We describe BSL and illustrate it on an example analyzed with Bandera and the Spin model checker. (24 Refs)
Subfile: C
Descriptors: finite state machines; formal specification; Java; multi-threading; program verification; specification languages
Identifiers: language framework; checkable properties; dynamic software; correctness properties; software systems; model checking; finite state models; program source code; abstracting; tractable checking; program properties; **assertion** violations; general program properties; dynamic multi-threaded Java programs; Bandera system; Java code; Bandera Specification Language; BSL; general **assertions** ; pre/post conditions; program objects; object quantifiers; state/event sequencing properties; Spin model checker
Class Codes: C6150G (Diagnostic, testing, debugging and evaluating systems); C6110F (Formal methods); C4240 (Programming and algorithm theory); C4220 (Automata theory); C6110J (Object-oriented programming); C6140D (High level languages); C6150N (Distributed systems software); C6110P (Parallel programming)

9/5/14 (Item 2 from file: 2)
DIALOG(R)File 2:INSPEC

5985680 INSPEC Abstract Number: C9809-6110B-033
Title: Practical issues in the use of ABFT and a new failure model
Author(s): Silva, J.G.; Prata, P.; Rela, M.; Madeira, H.
Author Affiliation: Dept. de Engenharia Inf., Coimbra Univ., Portugal
Conference Title: Digest of Papers. Twenty-Eighth Annual International Symposium on Fault-Tolerant Computing (Cat. No.98CB36224) p.26-35

Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA
Publication Date: 1998 Country of Publication: USA xx+470 pp.
ISBN: 0 8186 8470 4 Material Identity Number: XX98-01770
U.S. Copyright Clearance Center Code: 0731-3071/98/$10.00
Conference Title: Proceedings of 28th International Symposium on Fault Tolerant Computing
Conference Sponsor: IEEE Comput. Soc. Tech. Committee on Fault-Tolerant Comput.; IFIP WG 10.4 on Dependable Comput. & Fault Tolerance
Conference Date: 23-25 June 1998 Conference Location: Munich, Germany
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P); Experimental (X)
Abstract: We study the behavior of algorithm based fault tolerance (ABFT) techniques under faults injected according to a quite general fault model. Besides the problem of roundoff error in floating point arithmetic we identify two further weakpoints, namely lack of protection of data during input and output, and incorrect execution of the correctness checks. We propose the robust ABFT technique to handle those weakpoints. We then generalize it to **programs** that use **assertions** , where similar **problems** arise, leading to the technique of robust **assertions** , whose effectiveness is shown by **fault** injection experiments on a realistic control **application** . With this technique a system follows a new **failure** model, that we call fail-bounded, where with high probability all results produced are either correct or, if wrong, they are within a certain bound of the **correct** value, whose exact value depends **on** the output **assertions** used. We claim that this failure model is very useful to describe the behavior of many low redundancy systems. (24 Refs)
Subfile: C
Descriptors: computerised control; floating point arithmetic; redundancy; roundoff errors; **software fault** tolerance
Identifiers: ABFT; failure model; algorithm based fault tolerance; fault model; roundoff error; floating point arithmetic; correctness check; robust **assertions** ; fault injection; control application; fail bounded model; probability; low redundancy systems
Class Codes: C6110B (Software engineering techniques); C4240 (Programming and algorithm theory)

9/5/15 (Item 3 from file: 2)
DIALOG(R)File 2:INSPEC

5827974 INSPEC Abstract Number: C9803-6110B-030
Title: On-line software error **detection by executable** assertions : **from theory to practice**
Author(s): Rabejac, C.
Author Affiliation: Matra Marconi, Toulouse, France
Conference Title: SAFECOMP 95. 14th International Conference on Computer Safety, Reliability and Security p.390-402
Editor(s): Rabe, G.
Publisher: Springer-Verlag, Berlin, Germany
Publication Date: 1995 Country of Publication: Germany xii+516 pp.
ISBN: 3 540 19962 4 Material Identity Number: XX96-00092
Conference Title: Proceedings of International Conference on Computer Safety, Reliability and Security - SAFECOMP '95
Conference Sponsor: Eur. Workshop on Ind. Comput. Syst. Tech. Committee 7 ; Eur. Commission-Joint Res. Centre-Inst. Styst. Eng. & Informatics; et al
Conference Date: 11-13 Oct. 1995 Conference Location: Belgirate, Italy
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: **Software faults** are a major concern for safety-critical systems, particularly because they cannot be entirely removed during testing and validation phases. Executable **assertions** , **inserted** in source **code** and checked throughout operational execution, can detect **errors** caused by these residual **software faults** . This paper describes a method to design efficient executable **assertions** for on-line **software error** detection. The proposed method includes mechanisms to check **software** execution through the two complementary aspects of data and

control-flow. We then apply this method to two experimental case studies, in order to validate the key concepts involved. In the first case, validation involves **software fault** injection techniques, whereas in the second case the formal method B is used. (11 Refs)
Subfile: C
Descriptors: data flow analysis; program testing; safety-critical software; **software fault** tolerance
Identifiers: online **software error** detection; executable **assertions** ; **software faults** ; safety-critical systems; validation; testing; residual **software faults** ; control-flow; data flow analysis; **software fault** injection; formal method B; fault-tolerance techniques
Class Codes: C6110B (Software engineering techniques); C6150G (Diagnostic, testing, debugging and evaluating systems)

9/5/16 (Item 4 from file: 2)
DIALOG(R)File 2:INSPEC

5769054 INSPEC Abstract Number: C9801-6150G-019
Title: Improve your programming with asserts
Author(s): Rosenblum, B.D.
Journal: Dr. Dobb's Journal vol.22, no.12 p.60, 62-3
Publisher: Miller Freeman,
Publication Date: Dec. 1997 Country of Publication: USA
CODEN: DDJSDM ISSN: 1044-789X
SICI: 1044-789X(199712)22:12L.60:IYPW;1-G
Material Identity Number: B719-97011
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: Effective use of asserts will help you track and **fix** bugs faster, keep your project **on** schedule, and produce a bug-free application. Because they slow execution, however, it is important to maintain 2 versions of your application during the entire development cycle, one with asserts, for debugging, and one without, for the customers. Users of your application never see the debug version that you and your quality-assurance engineers have tested, but they will appreciate the added stability from the correct use of asserts. The author offers guidance on their use; one should **assert** explicit programmer **errors** , public API functions, assumptions, reasonable limits, unimplemented and untested **code** , and classes; one should not **assert** memory **errors** , resource **failures** , or FALSE; and one should not use VERIFY. An **assert** that merely returns the message " **assert** " is of value, but the author lists what should be stated in an **assert** to optimise its value. (0 Refs)
Subfile: C
Descriptors: program debugging; software engineering
Identifiers: asserts; bug tracking; bug-free application; explicit programmer errors; public API functions; assumptions; reasonable limits; unimplemented code; untested code; classes; memory errors; resource failures; FALSE; VERIFY
Class Codes: C6150G (Diagnostic, testing, debugging and evaluating systems); C6110B (Software engineering techniques)

9/5/18 (Item 6 from file: 2)
DIALOG(R)File 2:INSPEC

00912055 INSPEC Abstract Number: C76014016
Title: Towards correctness proofs for command programs and job executions
Author(s): Neuhold, E.J.
Author Affiliation: Univ. of Stuttgart, Stuttgart, West Germany
Conference Title: Computer Science Conference /sup '/75. (Abstracts only received) p.48
Publisher: ACM, New York, NY, USA
Publication Date: 1975 Country of Publication: USA xxiv+63 pp.

Conference Sponsor: ACM
Conference Date: 18-20 Feb. 1975 Conference Location: Washington, DC, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Theoretical (T)
Abstract: Research **on** program **correctness** mostly has ignored the area of proving the correctness of complete computer jobs, encompassing command programs, compilers, linkage-editors, loaders and user **programs** . A considerable part of all jobs **fail** not because of user **program failure** but for **errors** in the command **programs** . Only recently attention has been focused **on** proving the **correctness** of command programs. To extend the correctness proof for a command program to a complete job it becomes necessary to associate reasonable input and output **assertions** with the different job components and to relate them to the sometimes quite different **assertions** required for the correctness proofs of command programs. It becomes especially important to combine the deterministic proving techniques developed for the command program area with the heuristic methods required for the other more complex components of a computer job. (1 Refs)
Subfile: C
Descriptors: supervisory and executive programs
Identifiers: correctness proofs; command programs; job executions
Class Codes: C4290 (Other computer theory); C6150J (Operating systems)

9/5/19 (Item 1 from file: 233)
DIALOG(R)File 233:Internet & Personal Comp. Abs.

00580062 00DD03-001
White-box testing -- White-box testing should check every line of code
Cole, Oliver
Dr. Dobb's Journal , March 1, 2000 , v25 n3 p23-28, 5 Page(s)
ISSN: 1044-789X
Languages: English
Document Type: Articles, News & Columns
Geographic Location: United States
Focuses on white-box, or structural, software testing, in addition to black-box, or functional testing. Indicates that white-box testing (WBT) strategies include designing tests wherein every line of source code is executed at least once, or every function is required to be individually tested. States that various testing tools let one perform WBT on executables without the need for an interactive debugger, which allows for the best use of a test environment, and the benefits of testing the actual executable that will be delivered. Notes that WBT requires visibility into the executable to determine what to test, and a method to determine the outcome of the test. Attention is given to using a precise clock to perform WBT of timing; WBT's simplifying of **fault** injection into **programs** ; WBT advantages with regard to test coverage; and **adding assertions** to **code** to increase **software** quality. Includes seven code listings. (jon)
Descriptors: Testing; Benchmark Testing; Software; Application Development; Product Development; Quality Control

9/5/24 (Item 4 from file: 6)
DIALOG(R)File 6:NTIS

1245182 NTIS Accession Number: N86-23321/0
Dynamic Assertion **Testing of Flight Control Software**
Andrews, D. M. ; Mahmood, A. ; Mccluskey, E. J.
Stanford Univ., CA.
Corp. Source Codes: 009225000; S0380476
Sponsor: National Aeronautics and Space Administration, Washington, DC.
Report No.: NAS 1.26:176715; SU-HICSS-19; NASA-CR-176715
Jul 85 25p
Languages: English
Journal Announcement: GRAI8616; STAR2413

Order this product from NTIS by: phone at 1-800-553-NTIS (U.S. customers); (703)605-6000 (other countries); fax at (703)321-8547; and email at orders@ntis.fedworld.gov. NTIS is located at 5285 Port Royal Road, Springfield, VA, 22161, USA.
NTIS Prices: PC A02/MF A01
Country of Publication: United States
Contract No.: NAG2-246
Digital Flight Control System (DFCS) software was used as a test case for **assertion** testing. The **assertions** were written and embedded in the **code** , then **errors** were **inserted** (seeded) one at a time and the **code** executed. Results indicate that **assertion** testing is an effective and efficient method of detecting **errors** in flight **software** . Most **errors** are eliminate at an earlier stage in the development than before.
Descriptors: **Fault** tolerance; *Flight control; Dynamic tests; **Error** analysis; **Software** tools; Data flow analysis; Flight simulation; Format; On-line systems; Redundancy
Identifiers: NTISNASA
Section Headings: 51E (Aeronautics and Aerodynamics--Avionics); 62GE (Computers, Control, and Information Theory--General)

9/5/25 (Item 5 from file: 6)
DIALOG(R)File 6:NTIS

0965307 NTIS Accession Number: N82-22909/7/XAB
A Formalized Proof System for Total Correctness of While Programs
Bergstra, J. A. ; Klop, J. W.
Mathematisch Centrum, Amsterdam (Netherlands).
Corp. Source Codes: 017407000; S2885898
Sponsor: National Aeronautics and Space Administration, Washington, DC.
Report No.: MC-IW-175/81
Oct 81 19p
Languages: English
Journal Announcement: GRAI8218; STAR2013
Submitted for Publication.
Order this product from NTIS by: phone at 1-800-553-NTIS (U.S. customers); (703)605-6000 (other countries); fax at (703)321-8547; and email at orders@ntis.fedworld.gov. NTIS is located at 5285 Port Royal Road, Springfield, VA, 22161, USA.
NTIS Prices: PC A02/MF A01
Country of Publication: Netherlands
Data specifications based on schemes which are a slight generalization of first order specifications are introduced. For a schematic specification (Sigma, the set of all E), Hoare's Logic HL (Sigma, the set of all E) for partial correctness is defined as usual. A proof system (Sigma, the set of all E) **assertion** sign p tends to the limit S tends down to the limit for termination **assertions** is defined. Completeness with reference to second order semantics is proven. A translation of a standard proof system HL sub T(A) for total **correctness on** a structure A into this format is provided.
Descriptors: **Error** detection codes; *Formalism; *Mathematical logic; * **Program** verification (Computers); Axioms; Completeness; Semantics; Set theory
Identifiers: *Foreign technology; NTISNASAE
Section Headings: 62B (Computers, Control, and Information Theory--Computer Software)

9/5/26 (Item 6 from file: 6)
DIALOG(R)File 6:NTIS

0682003 NTIS Accession Number: AD-A050 154/4/XAB
The Semiautomatic Generation of Inductive Assertions **for Proving Program Correctness**
(Interim rept. 1 Jul 74-30 Jun 77)
Elspas, B. ; Boyer, R. S. ; Levitt, K. N. ; Moore, J. S. ; Robinson, L.

SRI International Menlo Park Calif
Corp. Source Codes: 410281
Report No.: AFOSR-TR-78-0114
Nov 77 126p
Journal Announcement: GRAI7810
Order this product from NTIS by: phone at 1-800-553-NTIS (U.S. customers); (703)605-6000 (other countries); fax at (703)321-8547; and email at orders@ntis.fedworld.gov. NTIS is located at 5285 Port Royal Road, Springfield, VA, 22161, USA.
NTIS Prices: PC A07/MF A01
Contract No.: F44620-73-C-0068; 2304; A2
This interim report describes progress **on** a project aimed at **solving** a serious **problem** that has been encountered in attempts to make **program** correctness proving a practical technique for **software** verification. The principal **problem** addressed here is the difficulty of synthesizing so-called loop **assertions** in connection with the main method now under study for program proving. Several rather diverse approaches, some of them constituting such alternatives to the present technique, are considered here: transformation of programs into primitive recursive form before verification, the method of generator induction for proof of properties of complex data structures, the use of a hierarchical design methodology to structure programs so as to minimize the need for loop **assertions** , and methods related to subgoal induction and computational induction. The two latter methods were analyzed in detail and compared with the present approach to arrive at a better understanding of their mutual relationships.
Descriptors: *Computer program verification; *Computer program reliability; Computer programming; Computer architecture; Hierarchies; Decision making; Algorithms; Closed loop systems; Finite difference theory; Recursive functions; Heuristic methods; Flow charting
Identifiers: NTISDODXA
Section Headings: 62B (Computers, Control, and Information Theory--Computer Software)
?

17/5/1 (Item 1 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

05383244 E.I. No: EIP99104832976
Title: Verification of control flow based security properties
Author: Jensen, T.; Le Metayer, D.; Thorn, T.
Corporate Source: IRISA/CNRS/INRIA, Rennes, Fr
Conference Title: Proceedings of the 1999 IEEE Symposium on Security and Privacy
Conference Location: Oakland, CA, USA Conference Date: 19990509-19990512
Sponsor: IEEE Computer Society
E.I. Conference No.: 55369
Source: Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy 1999. p 89-103
Publication Year: 1999
CODEN: PSSPEO ISSN: 1063-7109
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 9911W3
Abstract: A fundamental **problem** in **software** -based security is whether local security checks **inserted** into the **code** are sufficient to implement a global security property. We introduce a formalism based on a two-level linear-time temporal logic for specifying global security properties pertaining to the control-flow of the program, and illustrate its expressive power with a number of existing properties. We define a minimalistic, security-dedicated program model that only contains procedure call and run-time security checks and propose an automatic method for verifying that an implementation using local security checks satisfies a global security property. For a given formula in the temporal logic we prove that there exists a bound on the size of the states that have to be considered in order to assure the validity of the formula: this reduces the **problem** to finite-state model **checking** . Finally, we instantiate the framework to the security architecture proposed for Java (JDK 1.2). (Author abstract) 33 Refs.
Descriptors: *Security of data; Software engineering; Formal logic; Computer programming; Mathematical models; Java programming language
Identifiers: Global security property; Control flow verification; Temporal logic; Run-time security checks
Classification Codes:
723.1.1 (Computer Programming Languages)
723.2 (Data Processing); 723.1 (Computer Programming); 721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory); 921.6 (Numerical Methods)
723 (Computer Software); 721 (Computer Circuits & Logic Elements); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

17/5/3 (Item 3 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

04485317 E.I. No: EIP96083301200
Title: Compiler-assisted generation of error **-detecting parallel** programs
Author: Roy-Chowdhury, A.; Banerjee, P.
Corporate Source: IBM T. J. Watson Research Cent, Yorktown Heights, NY, USA
Conference Title: Proceedings of the 1996 26th International Symposium on Fault-Tolerant Computing
Conference Location: Sendai, Jpn Conference Date: 19960625-19960627
Sponsor: IEEE
E.I. Conference No.: 45241
Source: Proceedings - Annual International Conference on Fault-Tolerant Computing 1996. IEEE, Los Alamitos, CA, USA,96CB35969. p 360-369
Publication Year: 1996

CODEN: PFTCDY ISSN: 0791-3071
Language: English
Document Type: CA; (Conference Article) Treatment: G; (General Review); T; (Theoretical); X; (Experimental)
Journal Announcement: 9610W4
Abstract: We have developed an automated, compile time approach to generating **error** -detecting parallel **programs** . The compiler is used to identify statements implementing affine transformations within the program and automatically **insert code** for computing, manipulating, and comparing checksums in order to detect data errors at runtime. Statements which do not implement affine transformations are checked by duplication. Checksums are reused from one loop to the next if this is possible, rather than recomputing checksums for every statement. A global dataflow analysis is performed in order to determine points at which checksums need to be recomputed. We also use a novel method of specifying the data distributions of the check data using data distribution directives so that the computations on the original data and the corresponding check computations are performed on different processors. Results on the time overhead and **error** coverage of the **error** detecting parallel **programs** over the original **programs** are presented on an Intel Paragon distributed memory multicomputer. (Author abstract) 20 Refs.
Descriptors: Parallel processing systems; **Error** detection; **Program** compilers; Parallel algorithms; **Fault** tolerant computer systems; Encoding (symbols); Data handling; Data reduction; Computational complexity; Time sharing programs
Identifiers: **Check** sum encoding; Compiler assisted **fault** tolerance; **Error** detecting parallel **programs**
Classification Codes:
722.4 (Digital Computers & Systems); 721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory); 723.1 (Computer Programming); 723.2 (Data Processing)
722 (Computer Hardware); 721 (Computer Circuits & Logic Elements); 723 (Computer Software); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

17/5/19 (Item 2 from file: 2)
DIALOG(R)File 2:INSPEC

6539350 INSPEC Abstract Number: C2000-05-6150C-002
Title: Detection of run-time computer errors **, by provision of special** code **in compilers**
Author(s): Loveless, T.
Journal: Elektronik vol.49, no.3 p.140-5
Publisher: WEKA-Fachzeitschriften,
Publication Date: 8 Feb. 2000 Country of Publication: Germany
CODEN: EKRKAR ISSN: 0013-5658
SICI: 0013-5658(20000208)49:3L.140:DTCE;1-D
Material Identity Number: E071-2000-004
Language: German Document Type: Journal Paper (JP)
Treatment: Practical (P); Experimental (X)
Abstract: Discusses problems of detecting dynamic run-time **errors** in embedded computer systems, and lists **code** assembler **code** extracts for **insertion** in compilers, which are to act as "listening posts" for faults. Errors to be detected are stated to include global variables, local variables, stack over-runs, pointer references, arithmetic and increment/decrement pointers and standard functions such as MEMSET and BCOPY. Operation of an **error - checker** is described. **Testing** of **error** detectors against intentional memory blocking, is reported. It is concluded that **code** quality can be improved by planned **error checking** . (0 Refs)
Subfile: C
Descriptors: embedded systems; error detection codes; program compilers; program testing
Identifiers: run-time computer errors; compilers; dynamic run-time errors ; embedded computer; code assembler code extracts; listening posts; global variables; local variables; stack over-runs; pointer references; arithmetic

pointers; increment/decrement pointers; standard functions; MEMSET; BCOPY; **error - checker** ; error detectors; memory blocking; code quality; planned **error checking**
Class Codes: C6150C (Compilers, interpreters and other processors); C6150G (Diagnostic, testing, debugging and evaluating systems)

17/5/24 (Item 7 from file: 2)
DIALOG(R)File 2:INSPEC

4509292 INSPEC Abstract Number: B9312-1265F-014, C9312-5470-009
Title: Two software **techniques for on-line** error **detection**
Author(s): Miremadi, G.; Karlsson, J.; Gunneflo, U.; Torin, J.
Author Affiliation: Dept. of Comput. Eng., Chalmers Univ. of Technol., Goteborg, Sweden
Conference Title: Digest of Papers. The 1992 IEEE Workshop on Fault-Tolerant Parallel and Distributed Systems (Cat. No.92TH0449-9) p. 328-35
Publisher: IEEE Comput. Soc. Press, Los Alamitos, CA, USA
Publication Date: 1992 Country of Publication: USA viii+233 pp.
ISBN: 0 8186 2870 7
U.S. Copyright Clearance Center Code: 0 8186 2870 7/92$03.00
Conference Sponsor: IEEE
Conference Date: 6-7 July 1992 Conference Location: Amherst, MA, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: Two **software** based techniques for on-line detection of control flow **errors** are presented and evaluated by fault injection. One technique, called block signature self checking (BSSC), checks the control flow between **program** blocks. The other, called error capturing **instructions** (ECIs), **inserts** ECIs (e.g. trap **instructions**) in the **program** area, the data area and the unused area of the memory. To demonstrate these techniques, a program has been developed which modifies the executable **code** for the MC6809E 8-bit microprocessor. The **error** detection techniques were evaluated using two fault injection techniques: heavy-ion radiation from a Californium-252 source and power supply disturbances. Combinations of the two **error** detection techniques were **tested** for three different workloads. A combination BSSC, ECIs and a watchdog timer was also evaluated. (19 Refs)
Subfile: B C
Descriptors: computer testing; error detection; fault tolerant computing; logic testing
Identifiers: online error detection; software techniques; control flow errors; fault injection; block signature self checking; **error** capturing **instructions** ; executable code; MC6809E 8-bit microprocessor; heavy-ion radiation; Californium-252 source; power supply disturbances; watchdog timer
Class Codes: B1265F (Microprocessors and microcomputers); B1265B (Logic circuits); C5470 (Performance evaluation and testing); C5210 (Logic design methods)

17/5/25 (Item 8 from file: 2)
DIALOG(R)File 2:INSPEC

04372457 INSPEC Abstract Number: C9305-6150G-002
Title: Testing programs **to detect malicious** faults
Author(s): Hamlet, R.
Author Affiliation: Dept. of Comput. Sci., Portland State Univ., OR, USA
Conference Title: Dependable Computing for Critical Applications 2 p. 375-92
Editor(s): Meyer, J.F.; Schlichting, R.D.
Publisher: Springer-Verlag, Wien, Austria
Publication Date: 1992 Country of Publication: Austria xiii+437 pp.
ISBN: 3 211 82330 1

Conference Sponsor: IFIP; IEEE; EWICS; Univ. Arizona
Conference Date: 18-20 Feb. 1991 Conference Location: Tucson, AZ, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: **Program** testing has traditionally been of two kinds: for **fault** finding (debugging), and for establishing operational reliability (confidence). The author investigates the question of using traditional methods to determine the dependability of a **program** , under two assumptions: (1) the only sources of failure are inadvertent mistakes in design, coding, etc., and the **program** developers cooperate in trying to eliminate such **faults** . (2) the source of **failure** is sabotage-malicious **code** is **inserted** in the **program** and cleverly concealed. Paradoxically, it appears to be easier to detect sabotage than subtle unintentional mistakes, in the off-line situation where the sabotage takes place during development, and must be detected prior to program release. Furthermore, the very situations that can make traditional testing a nightmare, for example, real-time constraints, actually may help a tester trying to detect sabotage. (22 Refs)
Subfile: C
Descriptors: program testing; security of data
Identifiers: program testing; malicious faults detection; debugging; operational reliability; dependability; real-time constraints
Class Codes: C6150G (Diagnostic, testing, debugging and evaluating systems); C6130S (Data security)

17/5/26 (Item 9 from file: 2)
DIALOG(R)File 2:INSPEC

04286470 INSPEC Abstract Number: C9301-6150G-004
Title: Purify: fast detection of memory leaks and access errors
Author(s): Hastings, R.; Joyce, B.
Conference Title: Proceedings of the Winter 1992 USENIX Conference p. 125-36
Publisher: USENIX, Berkeley, CA, USA
Publication Date: 1991 Country of Publication: USA viii+451 pp.
Conference Sponsor: USENIX
Conference Date: 20-24 Jan. 1992 Conference Location: San Francisco, CA, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: This paper describes Purify, a software testing and quality assurance tool that detects memory leaks and access **errors** . Purify **inserts** additional **checking instructions** directly into the object **code** produced by existing compilers. These instructions check every memory read and write performed by the **program** -under-test and detect several types of access **errors** , such as reading uninitialized memory or writing to freed memory. Purify inserts checking logic into all of the code in a program, including third-party and vendor object-code libraries, and verifies system call interfaces. In addition, Purify tracks memory usage and identifies individual memory leaks using a novel adaptation of garbage collection techniques. Purify produces standard executable files compatible with existing debuggers, and runs a Sun Microsystems' SPARC family of workstations. Purify's nearly-comprehensive memory access checking slows the target program down typically by less than a factor of three and has resulted in significantly more reliable software for several development groups. (6 Refs)
Subfile: C
Descriptors: program testing; quality control; software tools; storage management
Identifiers: Sun Microsystem; Purify; software testing; quality assurance tool; memory leaks; access errors; compilers; object-code libraries; system call interfaces; garbage collection techniques; debuggers; SPARC family
Class Codes: C6150G (Diagnostic, testing, debugging and evaluating systems); C6115 (Programming support); C6120 (File organisation)

17/5/62 (Item 1 from file: 95)
DIALOG(R)File 95:TEME-Technology & Management

00911317 E95086655080
HotWire - a visual debugger for C++
(HotWire, ein visueller Debugger fuer C++)
Laffra, C; Malhotra, A
IBM Res. Center, Yorktown Heights, USA
Proc. of the 1994 USENIX C++ Conf., Cambridge, USA, Apr 11-14, 19941994
Document type: Conference paper Language: English
Record type: Abstract
ISBN: 1-880446-60-X

ABSTRACT:
The authors argue that visualization is essential in a modern debugger. Instead of **inserting** debug statements throughout the **code** , it should be possible to easily define visualizations while running the program under control of the debugger, resulting in what might be called 'visual printf's'. A visualization of a C++ program can provide exciting insights. Bugs that cannot be found that easily with non-visual techniques are now found, just by watching the visualizations. However, the mechanisms to define the visualizations should be easy to understand, easy to apply and cause only minimal overhead to the programmer (who is the end-user of the visual debugger). HotWire is not only equipped with a couple of standard visualizations, but also with a small declarative script language (using constraints) that can be used to define new custom visualizations. This paper addresses user interface aspects of debugging tools. Specifically, the user interface of HotWire, a debugger for C++ and SmallTalk on AIX and OS/2 is described.

DESCRIPTORS: **ERROR** FINDING; PROGRAMMING AID; **TEST** AID **PROGRAM** ; SUPERVISORY **PROGRAMS** ; MONITORS--DISPLAY UNIT; GRAPHIC DATA OUTPUT; GRAPHIC DATA PROCESSING; DEFECT DETECTION; PROGRAM DEVELOPMENT; PROGRAMMING LANGUAGES; OBJECT ORIENTED PROGRAMMING; USER INTERFACES; WINDOW SYSTEM; USER FRIENDLINESS; MAN MACHINE SYSTEMS; SOFTWARE TOOLS; UNIX OPERATING SYSTEMS; C PLUS PLUS--PROGRAMMING LANGUAGE
IDENTIFIERS: VISUELLER DEBUGGER; visueller Debugger; C plus plus; Software-Werkzeug

17/5/63 (Item 2 from file: 95)
DIALOG(R)File 95:TEME-Technology & Management

00682274 I93057394927
Concurrent runtime monitoring of formally specified programs
(Simultane Ueberwachung von formal spezifizierten Programmen zur Ausfuehrungszeit)
Sankar, S; Mandal, M
Stanford Univ., CA, USA
Computer, Long Beach, v26, n3, pp32-41, 1993
Document type: journal article Language: English
Record type: Abstract
ISSN: 0018-9162

ABSTRACT:
A methodology for continuously monitoring a program for specification consistency during program execution is described. Prior development of the formal specification and program is assumed. The program is annotated with constructs from a formal specification language, and the formal specification constructs are transformed into checking **code** , which is then **inserted** into the underlying **program** . Calls to this checking **code** are **inserted** into underlying **program** wherever it can potentially become inconsistent with its specification. If an inconsistency does in fact occur, diagnostic information is provided. The implementation of such a system for Anna (annotated Ada) subtype annotations is presented.

DESCRIPTORS: FORMAL SPECIFICATION; PARALLEL PROGRAMMING; DESCRIPTION LANGUAGES; PROCESS MONITORING; **PROGRAM TESTING** ; **PROGRAM** VERIFICATION; DEFECT DETECTION; **ERROR** DIAGNOSIS; IMPLEMENTATION; COMPUTER **PROGRAM** ; ADA--PROGRAMMING LANGUAGE; CONCURRENT WORKING
IDENTIFIERS: SYSTEM MONITORING; SPECIFICATION CONSISTENCY; PROGRAM EXECUTION; FORMAL SPECIFICATION LANGUAGE; FORMAL SPECIFICATION CONSTRUCTS; CHECKING CODE; UNDERLYING PROGRAM; DIAGNOSTIC INFORMATION; ANNA; ANNOTATED ADA; SUBTYPE ANNOTATIONS; Programmpruefung; formale Programmspezifikation

17/5/64 (Item 3 from file: 95)

00675544 I93034772928
A design method for cost-effective self- testing checker **for optimal d-unidirectional** error **detecting codes**
(Eine Entwurfsmethode fuer kosteneffektive selbsttestende Pruefgeraete fuer optimale d-eingerichtete Fehlererkennungs-Codes)
Fujiwara, E; Yoshikawa, M
Fac. of Eng., Tokyo Inst. of Technol., Japan
IEICE Transactions on Information and Systems, vE75-D, n6, pp771-777, 1992
Document type: journal article Language: English
Record type: Abstract
ISSN: 0916-8532

ABSTRACT:
Unidirectional asymmetric terror control codes have extensively been studied, not only from theoretical interest but from application to computer systems or communication systems. Recently, attention has been focused on detecting only d, not all, unidirectional errors, that is, d bits unidirectional error detecting (d-UED) codes. Borden (1982) proposed an optimal nonsystematic d-UED code. The paper shows a design method for cost-effective self-testing checker for the optimal d-UED code. The checking policy is to check whether condition of the Borden code satisfies or not. The proposed checker includes the parallel weight counter, the comparator and the modulo adder in which the residue operation is defined and hence this makes the circuit self-testing. These circuits are designed to have all possible input patterns in order to satisfy the self-testing property. Finally, the proposed checker has greatly reduced hardware compared to the existing ones.

DESCRIPTORS: GRAND SCALE INTEGRATION; COMPARATORS--CIRCUITS; SELF TESTING; COMMUNICATION SYSTEMS; DEFECT DETECTION; CODES; **TEST** DEVICES; **ERROR** RESILIENT SCHEME; **ERROR** DETECTION CODES; INTEGRATED CIRCUIT **TESTING** ; INTEGRATED LOGIC CIRCUITS; LOGIC TESTING
IDENTIFIERS: BUILT IN SELF TEST; COST EFFECTIVE SELF **TESTING** CHECKER; OPTIMAL D UNIDIRECTIONAL **ERROR** DETECTING CODES; TERROR CONTROL CODES; OPTIMAL NONSYSTEMATIC D UED **CODE** ; CHECKING POLICY; BORDEN **CODE** ; PARALLEL WEIGHT COUNTER; MODULO **ADDER** ; RESIDUE OPERATION; VLSI-Schaltung; Selbsttest; Pruefgeraet
?

```
File 275:Gale Group Compu[illegible] DB(TM) 1983-2003/Jun 12
         (c) 2003 The Gale Group
File 621:Gale Group New Prod.Annou.(R) 1985-2003/Jun 11
         (c) 2003 The Gale Group
File 636:Gale Group Newsletter DB(TM) 1987-2003/Jun 10
         (c) 2003 The Gale Group
File  16:Gale Group PROMT(R) 1990-2003/Jun 12
         (c) 2003 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 148:Gale Group Trade & Industry DB 1976-2003/Jun 11
         (c)2003 The Gale Group
File 624:McGraw-Hill Publications 1985-2003/Jun 12
         (c) 2003 McGraw-Hill Co. Inc
File  15:ABI/Inform(R) 1971-2003/Jun 12
         (c) 2003 ProQuest Info&Learning
File 647:CMP  Computer Fulltext 1988-2003/May W3
         (c) 2003 CMP Media, LLC
File 674:Computer News Fulltext 1989-2003/Jun W2
         (c) 2003 IDG Communications
File 696:DIALOG Telecom. Newsletters 1995-2003/Jun 11
         (c) 2003 The Dialog Corp.
File 369:New Scientist 1994-2003/Jun W1
         (c) 2003 Reed Business Information Ltd.
File 112:UBM Industry News 1998-2003/Jun 12
         (c) 2003 United Business Media

Set     Items   Description
S1      98412   ASSERTION? ? OR ASSERT
S2     163722   (ERROR? ? OR FAIL? OR FAULT? ? OR PROBLEM? ? OR TROUBLE OR
              FLAW OR IRREGULAR? OR GLITCH?? OR FALSE OR ODD???? OR STRANGE-
              ???? OR ANOMAL? OR ABNORMAL?)(5N)(TEST??? OR CHECK???)
S3     188182   (ERROR? ? OR FAIL? OR FAULT? ? OR PROBLEM? ? OR TROUBLE OR
              FLAW OR IRREGULAR? OR GLITCH?? OR FALSE OR ODD???? OR STRANGE-
              ???? OR ANOMAL? OR ABNORMAL?)(5N)(CONDITION? ? OR STATE OR ST-
              ATES OR SITUATION OR STATUS)
S4     593362   (RECOVER? OR CORRECT? OR FIX??? OR MEND??? OR REMED??? OR -
              RECTIF? OR REPAIR? OR PATCH? OR RESTOR? OR RESOLV? OR SOLV?)(-
              5N)(ENABL? OR ON OR DISABL? OR OFF)
S5      53138   ((INSERT? OR PUT???? OR PLAC??? OR PLACEMENT OR ADD??? OR -
              APPEND? OR EMBED?)(5N)(CODE? ? OR INSTRUCTION? ? OR FUNCTION?
              ? OR COMMAND? ? OR ROUTINE? ? OR PROCEDURE? ?))(5W)(PROGRAM? ?
               OR CODE OR APPLICATION? ? OR SOFTWARE)
S6     610689   (ERROR? ? OR FAIL? OR FAULT? ? OR PROBLEM? ? OR TROUBLE OR
              FLAW OR IRREGULAR? OR GLITCH?? OR FALSE OR ODD???? OR STRANGE-
              ???? OR ANOMAL? OR ABNORMAL?)(10N)(PROGRAM? ? OR CODE OR APPL-
              ICATION? ? OR SOFTWARE OR INSTRUCTIONS OR OPERATIONS)
S7         19   S1:S3(S)S4(S)S5
S8         12   RD (unique items)
S9      79818   (WHEN OR IF OR SHOULD OR WHILE OR THAT)(5W)S4
S10        54   S9(S)S5
S11        37   RD (unique items)
S12        28   S11 NOT PY=2001:2003
S13       267   S1(S)(S9 OR S5)
S14        21   S1(S)(S9 OR S5)(S)S6
S15        17   RD (unique items)
```

8/3,K/1 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

02222385 SUPPLIER NUMBER: 21168440 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Is 2000 a budget victim?(John A Koskinen, Chmn, President's Council on Year 2000 Conversion) (Government Activity)
Dorobek, Christopher J.; Mayer, Merry
Government Computer News, v17, n31, p1(1)
Sept 21, 1998
ISSN: 0738-4300 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 1409 LINE COUNT: 00142

TEXT:
...and Horn, chairman of the House Government Oversight and Reform Subcommittee on Government Management, Information and Technology, handed out his latest grades for agency date **code** efforts. OMB has **added** the State Department to its list of six other agencies that it deems too far behind **on** systems **fixes** . The other agencies **on** the red-flag list are the Agency for International Development and the departments of Defense, Education, Energy, HHS and Transportation. State faces a significant challenge...

...of systems it is renovating nor did it report adequate progress on final validations of systems," OMB said. Dave Ames, deputy chief information officer in **State** 's Year 2000 **Problem** Program Management Office, said the department considers the year 2000 problem its No. 1 priority. The department will meet the goal of having systems ready...

8/3,K/2 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

02179629 SUPPLIER NUMBER: 20645192 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Mapping your business.(Visio Maps map database) (Software Review)(Evaluation)
Cunningham, Cliff
Computing Canada, v24, n20, p36(1)
May 25, 1998
DOCUMENT TYPE: Evaluation ISSN: 0319-0161 LANGUAGE: English
RECORD TYPE: Fulltext
WORD COUNT: 830 LINE COUNT: 00066

... a United States map easily (yes, Canadian maps are included). Graduated color or symbol maps are alternate options.
Two problems remained, however: of the 512 **places** I was importing by zip **code** , 51 were "not found." No further explanation was offered, no list of unknown zip codes was printed, and no method of **checking** these was available. The other **problem** was even more trying. While the dots were placed **correctly** , clicking **on** a dot elicited no information from the database. Knowing who or what a data point represents on the map would be most useful.
These problems...

8/3,K/3 (Item 3 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01507629 SUPPLIER NUMBER: 12011706 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Languages. (question-and-answer) (Column)
Terdeman, Sharon
PC Magazine, v11, n8, p427(2)
April 28, 1992
DOCUMENT TYPE: Column ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 1163 LINE COUNT: 00083

... number, plus the month number times 30 to calculate the day number of a date in the year. Using the byte-based array makes the **function** more compact while **adding** only a tiny piece of **code** .

A small **program** , DOW.C, is presented in Figure 2 to show off the dow function. The program takes a command line argument, such as 12 25 1992...

...for it. Then it calls the function and prints the day of the week that the date falls on. Note that, as written, DOW relies **on** the user for **correct** input. There's no **error checking** , and if, for example, you enter a day or month that's too large or if the format is incorrect (such as 12 25 92...

8/3,K/4 (Item 4 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01152573 SUPPLIER NUMBER: 00655556 (USE FORMAT 7 OR 9 FOR FULL TEXT)
A Smorgasbord of Fox & Geller Enhancements for dBASE III.
Hart, G.A.
PC Magazine, v4, n23, p219
Nov. 12, 1985
DOCUMENT TYPE: evaluaton ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 5021 LINE COUNT: 00383

... types of logic errors, primarily those involving misnesting of logical constructs or missing symbols. No such errors appeared in my existing program, but dUTIL worked **correctly on** a few **test** programs I wrote with deliberate **errors** . My **test** suite consisted of 21 programs totaling 107, 136 bytes (including comment headers and some **embedded** comments in the **code** itself). dUTIL combined these separate files into one 226,570-byte monster file, taking a 1/2 hour to do so on an AT.

The...

8/3,K/5 (Item 1 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

10415080 Supplier Number: 91213200 (USE FORMAT 7 FOR FULLTEXT)
Forward-error correction can enhance transmission capacity: forward-error correction can reduce bit-error rates and significantly extend transmission distances. The technique is finding increased use in long-haul systems. (Optical Networking).
Hecht, Jeff
Laser Focus World, v38, n8, p89(4)
August, 2002
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 1843

... Efforts to mitigate the effects of crosstalk noise pushed fiber-system developers to consider forward-error correction.

Error detection and correction

An error detection and **correction** code operates **on** a block of data. Its power depends on both the number of bits added to the block and the power of the mathematical ...that an error has occurred; it is harder to identify the incorrect bit and correct it. All codes have limited capacity, so large numbers of **errors** can block transmission.

Parity **checking** is a simple code that adds a parity bit to an eight-bit byte. The transmitter adds the data bits to determine if their sum...

8/3,K/6 (Item 1 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB

10343230 SUPPLIER NUMBER: 20949533 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Visio Maps leaves users lost. (new product from Visio Corp.) (Evaluation)
Cunningham, Cliff
Computer Dealer News, v14, n24, p40(1)
June 22, 1998
DOCUMENT TYPE: Evaluation ISSN: 1184-2369 LANGUAGE: English
RECORD TYPE: Fulltext
WORD COUNT: 747 LINE COUNT: 00059

... a United States map easily (yes, Canadian maps are included). Graduated color or symbol maps are alternate options.

Two problems remained, however. Of the 512 **places** I was importing by zip **code** , 51 were "not found." No further explanation was offered, no list of unknown zip codes was printed, and no method of **checking** these was available. The other **problem** was even more trying. While the dots were placed **correctly** , clicking **on** a dot elicited no information from the database. Knowing who or what a data point represents on the map would be most useful.

These problems...

8/3,K/7 (Item 2 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2003 The Gale Group. All rts. reserv.

10152015 SUPPLIER NUMBER: 20543096 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Alydaar Reports First Quarter Results: Revenues up Significantly and Strongly Profitable
PR Newswire, p0429CHW006
April 29, 1998
LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 683 LINE COUNT: 00069

... versus the "tools vendors" approach to remediating code. We have found that every customer's code has uniqueness, which makes it difficult to use an **off** -the-shelf tool to **fix** code. We believe that, based upon published data for our competitors and our internal tracking of results, the Company has the lowest error rate per...

...in performing unit testing of remediated code. Based upon our low error rate, some of our customers have eliminated the unit testing stage for repairing **code** to be **put** back into production. We believe that our factory approach, which enables us to handle large volumes of code in a short period of time, and our low **error** rate, which saves **testing** time and cost, will position us to be successful in winning contracts awards from large companies which have attempted to fix code internally and decide ...

8/3,K/8 (Item 3 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2003 The Gale Group. All rts. reserv.

03929893 SUPPLIER NUMBER: 07444772 (USE FORMAT 7 OR 9 FOR FULL TEXT)
The world economy, ten years from today. (20 Years: A Special 20th Anniversary Supplement)
Euromoney, pSS3(133)
June, 1989
ISSN: 0014-2433 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 70688 LINE COUNT: 06272

... checking the price on screen, the dealer would then have to do no more than confirm a trade by punching in his firm's transaction **code** .

The technology is available to do this: Nasdaq has had an automated execution system for five years, and the other major vendors, including Reuters and...can be fashioned, two major reforms are needed: a new price

system and convertibility of the rouble.

Cutting subsidies and allowing competitive prices puts companies **on** a proper business footing. Convertibility of the rouble is essential for foreign investors. "By the end of the 1990s these measures could well have been...now obliged to report on a Western accounting basis," says Newman.

As for people to put the plans into action, that shouldn't be a **problem** either. "Moscow Narodny and other **state** banks have been playing on Western stock exchanges for years.

Richard Evans ONE WORLD, ONE FUND

Does the world really need two organisations to manage...

8/3,K/9 (Item 4 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2003 The Gale Group. All rts. reserv.

03136536 SUPPLIER NUMBER: 05014609 (USE FORMAT 7 OR 9 FOR FULL TEXT)
The limits of software reliability.
Enfield, Ronald L.
Technology Review, v90, p36(7)
April, 1987
CODEN: TEREA ISSN: 0040-1692 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 3342 LINE COUNT: 00268

... lines of code: 1,000,000

Number of faults in the initial software (2% of the total, based on a widely reported average): 20,000

Faults remaining after **testing** (assuming that 90% of the **faults** are found and fixed): 2,000 Number of failures per year (10% of the faults, based **on** experience): 200 Faults **corrected** after failures: 200 Remaining faults: 1,800 Lines of **code added** or changed per year in **routine** maintenance (estimated at 10% per year): 100,000 Number of faults **added** to system (2% of new **code**): 2,000 Number of new faults remaining after debugging new code (assuming 90% of new faults removed): 200 Number of faults not discovered in previous...

8/3,K/10 (Item 1 from file: 674)
DIALOG(R)File 674:Computer News Fulltext
(c) 2003 IDG Communications. All rts. reserv.

084160
Saboteur guilty
Jury convicts net manager in landmark case.
Byline: Sharon Gaudin
Journal: Network World Page Number: 1
Publication Date: May 15, 2000
Word Count: 1699 Line Count: 150

Text:
... reformatted them, according to testimony at his trial for computer sabotage, which ended last week in a guilty verdict. And in a clever twist, Lloyd **added** a simple line of **code** to his time bomb so when that early-morning user logged on to the server, the screen flashed a message that said 'fixing.' Thus, no...

... manufacturing unsupervised, unprotected and unmaintained.And Lloyd told Network World that his attorneys have the missing programs. He said one of the defense's data **recovery** experts found them **on** a copy of the targeted file server. Lloyd's attorneys, however, did not present the programs or the expert during the trial."We got 'em...
... have the programs, which I do not believe they do, that wouldn't have changed the crime. The sabotage was still a crime."Lloyd's **assertion** is also at odds with expert testimony from Ontrack Data Services, a Minnesota company hired by Omega first to try to recover the programs and...

8/3,K/11 (Item 2 from file: 674)
DIALOG(R)File 674:Computer News Fulltext
(c) 2003 IDG Communications. All rts. reserv.

079225
Web monitoring tools help IT rest easier
Byline: SUZANNE GASPAR
Journal: Network World
Publication Date: November 08, 1999
Word Count: 855 Line Count: 78

Text:
... better way to monitor and manage the Web site.Sanders solved the problem by buying Platform Computing's SiteAssure, a Web monitoring tool that takes **corrective** action based **on** predefined policies. He uses it in conjunction with Freshwater Software's SiteScope real-time Web-server monitoring tool to keep the Web site up and...

... next stepWhen Sanders initially assessed the monitoring and management needs of his Web data center, he recognized its immaturity. He planned to first stabilize his **code** base and platform, then **add** products to take corrective action. Sanders sees Web management as a layering process: first encounter the problem and examine frequency, then look at the trend...

... does that job well on his network. He has put in the time and effort to tune SiteScope, configuring its rules to look for specific **error conditions** and various content strings. What's more, he can easily change the name of the IT staffer who is paged if a problem arises that...

8/3,K/12 (Item 1 from file: 696)
DIALOG(R)File 696:DIALOG Telecom. Newsletters
(c) 2003 The Dialog Corp. All rts. reserv.

00721921
GRAY MARKET PLAYSTATION2s ON RISE IN U.S.
CONSUMER MULTIMEDIA REPORT
April 17, 2000 DOCUMENT TYPE: NEWSLETTER
PUBLISHER: WARREN PUBLISHING INC.
LANGUAGE: ENGLISH WORD COUNT: 1079 RECORD TYPE: FULLTEXT

TEXT:
...consoles modified with $40 chip from playing counterfeit discs.

Beauty of system is that existing consoles need not be changed at all. Instead, Macrovision anticopy **code** is **embedded** on legitimate game discs and carries through when pirated disc is made on CD recorder. Inventor Roger Edwards says in patent that **putting code** on new game discs effectively retrofits "even the oldest consoles." CDilla tried to interest Sony but was rebuffed, said Macrovision U.K. Managing Dir. David...
...here syndrome," he told us.

PS software piracy is thorn in side of 3rd party developers and retailers, who repeatedly have petitioned Sony to address **situation** . **Problem** is most acute in Europe, where new game titles don't appear until long after release in Japan and U.S. By then, games already...

...but can't be played unless console is modified with chip that mimics legitimate authorization codes. Those are lost during copying process because CD recorders **correct** authentication **code** errors deliberately **placed on** legitimate titles to foil copying. Inexpensive mod-chip fools console into playing bogus game -- and

that's where Macrovision system foils pirates. Where legitimate disc...consoles modified with $40 chip from playing counterfeit discs.

Beauty of system is that existing consoles need not be changed at all. Instead, Macrovision anticopy **code** is **embedded** on legitimate game discs and carries through when pirated disc is made on CD recorder. Inventor Roger Edwards says in patent that **putting code** on new game discs effectively retrofits "even the oldest consoles." CDilla tried to interest Sony but was rebuffed, said Macrovision U.K. Managing Dir. David...

...here syndrome," he told us.

PS software piracy is thorn in side of 3rd party developers and retailers, who repeatedly have petitioned Sony to address **situation** . **Problem** is most acute in Europe, where new game titles don't appear until long after release in Japan and U.S. By then, games already...

...but can't be played unless console is modified with chip that mimics legitimate authorization codes. Those are lost during copying process because CD recorders **correct** authentication **code** errors deliberately **placed on** legitimate titles to foil copying. Inexpensive mod-chip fools console into playing bogus game -- and that's where Macrovision system foils pirates. Where legitimate disc...
?

12/3,K/1 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

02222385 SUPPLIER NUMBER: 21168440 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Is 2000 a budget victim?(John A Koskinen, Chmn, President's Council on Year 2000 Conversion) (Government Activity)
Dorobek, Christopher J.; Mayer, Merry
Government Computer News, v17, n31, p1(1)
Sept 21, 1998
ISSN: 0738-4300 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 1409 LINE COUNT: 00142

TEXT:
...and Horn, chairman of the House Government Oversight and Reform Subcommittee on Government Management, Information and Technology, handed out his latest grades for agency date **code** efforts. OMB has **added** the State Department to its list of six other agencies **that** it deems too far behind **on** systems **fixes** . The other agencies **on** the red-flag list are the Agency for International Development and the departments of Defense, Education, Energy, HHS and Transportation. State faces a significant challenge...

12/3,K/2 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

02179629 SUPPLIER NUMBER: 20645192 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Mapping your business.(Visio Maps map database) (Software Review)(Evaluation)
Cunningham, Cliff
Computing Canada, v24, n20, p36(1)
May 25, 1998
DOCUMENT TYPE: Evaluation ISSN: 0319-0161 LANGUAGE: English
RECORD TYPE: Fulltext
WORD COUNT: 830 LINE COUNT: 00066

... a United States map easily (yes, Canadian maps are included). Graduated color or symbol maps are alternate options.
Two problems remained, however: of the 512 **places** I was importing by zip **code** , 51 were "not found." No further explanation was offered, no list of unknown zip codes was printed, and no method of checking these was available. The other problem was even more trying. **While** the dots were placed **correctly** , clicking **on** a dot elicited no information from the database. Knowing who or what a data point represents on the map would be most useful.
These problems...

12/3,K/3 (Item 3 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01956614 SUPPLIER NUMBER: 18478311
A river runs through NT. (Novell's Green River NetWare network operating system upgrade will decrease number of users moving to Windows NT) (includes related article on Green River features) (Product Development)
DiDio, Laura
Computerworld, v30, n26, p1(2)
June 24, 1996
ISSN: 0010-4841 LANGUAGE: English RECORD TYPE: Abstract

...ABSTRACT: enhancements include embedded symmetrical multiprocessing that scales up to eight processors, a Netware Licensing Services facility that enables managers to simply click and type in **code** to delete, **add** or move licensed users on to the network and a crash **recovery** feature **that** will **enable** the NetWare file server self-diagnose and recover from

server crashes. In addition, Green River will feature increased volume capacity to manage up to 16...

12/3,K/4 (Item 4 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01778969 SUPPLIER NUMBER: 16891112 (USE FORMAT 7 OR 9 FOR FULL TEXT)
VR improves Motorola training program. (virtual reality used in robotic assembly line instruction)
Adams, Nina; Lang, Laura
AI Expert, v10, n5, p13(2)
May, 1995
ISSN: 0888-3785 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1113 LINE COUNT: 00091

... etches an identification number on each product. The entire line is controlled by an integrated factory control system that is operated by Motorola associates.
Programming **code** was written to **add** movement so the model would react **correctly** . **When** students turn **on** a power switch, the corresponding lights or equipment is powered. If the power hasn't been turned on and the students try to start the...

12/3,K/5 (Item 5 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01668763 SUPPLIER NUMBER: 15047570 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Peering inside the PE: a tour of the Win32 portable executable file format. (includes related articles on header formats, directory entries, header fields) (Tutorial)
Pietrek, Matt
Microsoft Systems Journal, v9, n3, p15(18)
March, 1994
DOCUMENT TYPE: Tutorial ISSN: 0889-9932 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 9524 LINE COUNT: 00734

... When the linker creates an EXE file, it makes an assumption about where the file will be mapped into memory. Based on this, the linker **puts** the real addresses of **code** and data items into the executable file. If for whatever reason the executable endsw up being loaded somewhere else in the virtual address space, the...

...into the image are wrong. The information stored in the .reloc section allows the PE loader to fix these addresses in the loaded image so **that** they're **correct** again. **On** the other hand, if the loader was able to load the file at the base address assumed by the linker, the .reloc section data isn...

12/3,K/6 (Item 6 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01507629 SUPPLIER NUMBER: 12011706 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Languages. (question-and-answer) (Column)
Terdeman, Sharon
PC Magazine, v11, n8, p427(2)
April 28, 1992
DOCUMENT TYPE: Column ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 1163 LINE COUNT: 00083

... number, plus the month number times 30 to calculate the day number

of a date in the year. Using the byte-based array makes the **function** more compact while **adding** only a tiny piece of **code** .

A small **program** , DOW.C, is presented in Figure 2 to show off the dow function. The program takes a command line argument, such as 12 25 1992...

...supplied, the program prompts you for it. Then it calls the function and prints the day of the week that the date falls on. Note **that** , as written, DOW relies **on** the user for **correct** input. There's no error checking, and if, for example, you enter a day or month that's too large or if the format is...

12/3,K/7 (Item 7 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01353003 SUPPLIER NUMBER: 08290014 (USE FORMAT 7 OR 9 FOR FULL TEXT)
TRACER: a debugging tool for OS/2 Presentation Manager development.
Hildebrand, Daniel
Microsoft Systems Journal, v5, n2, p63(11)
March, 1990
ISSN: 0889-9932 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2673 LINE COUNT: 00207

... The second, "development" debugging, is used during the evolution of code to monitor progress and prevent potential bugs. Commonly, a series of trace statements is **placed** within the **code** to monitor the values of selected variables, ensure that array boundaries are respected, verify **that** return codes are **correct** , check program states, and so **on** . These trace mechanisms usually redirect formatted output to devices such as files, printers, or auxiliary screens. Usually bugs are caught before they happen and when...

12/3,K/8 (Item 1 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

02212164 Supplier Number: 56905346 (USE FORMAT 7 FOR FULLTEXT)
Formal Systems America Awarded Two Year 2000 Contracts.
PR Newswire, p9164
Oct 26, 1999
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 753

... year.
Fixing the Fix
Due to time constraints, the majority of organizations chose to temporarily fix their systems by inserting a workaround called Windowing. Windowing **inserts** lines of **code** before each short date to temporarily expand it so **that** logical operations can be handled **correctly** .
Based **on** a review of over 200 million lines of NATURAL code, every 10th line of source code contains a date. For each date that was windowed, approximately five lines of **code** were **added** , thereby increasing the line count by approximately 50%. This substantial increase in the number of lines is likely to result in slower systems, potentially affecting...

12/3,K/9 (Item 2 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01691557 Supplier Number: 50237776 (USE FORMAT 7 FOR FULLTEXT)
TRW Builds and Validates High Performance RH32 Microprocessor for U.S. Air Force.
Business Wire, p08120099
August 12, 1998

Language: English Record Type: Fulltext
Article Type: Article
Document Type: Newswire; Trade
Word Count: 624

... satellite communications, wideband communications systems, next-generation surveillance and missile warning systems, and strategic missile systems.

The TRW RH32 includes advanced built-in fault tolerance **that enables** fast **recovery** and mission autonomy by utilizing hardware level instruction rollback and retry. The design has built-in **functions** to allow **embedded** real-time **software** debug and monitoring.

TRW has proven the design's technology independence by fabricating it at multiple foundries. This enables the porting of the TRW RH32...

12/3,K/10 (Item 1 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

04467400 Supplier Number: 56897237 (USE FORMAT 7 FOR FULLTEXT)
TELEPHONY.
Communications Daily, v19, n205, pNA
Oct 25, 1999
Language: English Record Type: Fulltext
Document Type: Newsletter; Trade
Word Count: 3227

... intended. State members are S.D. PSC Comr. Laska Schoenfelder, Tex. PUC Chmn. Pat Wood, Mo. Consumer Counsel Martha Hogerty. ------

FCC late Thurs. released order **that resolves** outstanding issues **on** telephone number administration. ...competed in old area code. (2) Clarified that states can allow callers to dial national 555 numbers using only 7 digits, even if call is **placed** from area **code** subject to overlay. (3) Said LECs can't assess fees for opening central office codes. (4) Gave states discretion to allow existing

12/3,K/11 (Item 2 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

03544108 Supplier Number: 47326384 (USE FORMAT 7 FOR FULLTEXT)
NATO: SFOR press briefing-Part 2
M2 Presswire, pN/A
April 25, 1997
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 2794

(USE FORMAT 7 FOR FULLTEXT)
TEXT:
...say, Prijedor are on the same code whereas towns in between are not. This web system exactly follows the IEBL boundary. It allows the Serbs, **if** I understand it **correctly** , to switch **off** all the connections between Republika Srpska and **places** outside of their **code** prefix area, if I'm correct. I'm just wondering why people gave into that demand. Why not go through Sarajevo? It would have been...

12/3,K/12 (Item 1 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

07035419 Supplier Number: 59116218 (USE FORMAT 7 FOR FULLTEXT)
Newswatch: Y2K Pessimists Face Post-Party Hangover.(Brief Article)
Marer, Eva
Financial Planning, pITEM00032007

Feb 1, 2000
Language: English Record Type: Fulltext
Article Type: Brief Article
Document Type: Magazine/Journal; Trade
Word Count: 670

... a proactive approach, for example, by updating computer systems, holding Y2K seminars and meeting individually with clients. "The Y2K bug taught us that if a **code** problem resides on **embedded** chips or hard drives, it can be very difficult and costly to locate and **fix** . But **if** your code resides **on** the Internet, you only have to go one place to fix it." Morrow anticipates more Internet-based computer applications and success for the companies developing...

12/3,K/13 (Item 2 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

06410870 Supplier Number: 54876835 (USE FORMAT 7 FOR FULLTEXT)
Eye on Washington: IT &the Feds:The Five Issues; Here are five areas where government policy will impact corporate IT -- and three people who could make a difference.(Company Business and Marketing)
Anthes, Patrick; Thibodeau, Gary H.
Computerworld, p52(1)
June 14, 1999
Language: English Record Type: Fulltext Abstract
Document Type: Magazine/Journal; Tabloid; Trade
Word Count: 1777

(USE FORMAT 7 FOR FULLTEXT)
TEXT:
...for companies to aggressively prepare for the date change. "This is a selfish and shortsighted position because it will not fix a single line of **code** , repair a single **embedded** chip or improve any contingency plans," he says.While industry as a whole is behind the bills, individual IT managers are split. At a recent...new guidelines for how to behave," says William Kovacic, a law professor at George Washington University in Washington.The case could have a dramatic impact **on** IT departments. **If** Microsoft loses, **remedies** may be imposed that could open up the Windows source code, break up the company or encourage computer vendors to offer alternative operating systems. If...

12/3,K/14 (Item 3 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

01074441 Supplier Number: 41194969 (USE FORMAT 7 FOR FULLTEXT)
PCA Ecko/Kaiser Facilitates Recycling
Distributor Sales, p30
March, 1990
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 92

(USE FORMAT 7 FOR FULLTEXT)
TEXT:
...of the Plastic Industry's plastic container coding system. The SPI coding system was developed to aid recyclers in sorting plastic containers by resin composition. **When** manufacturers **place** the **correct code on** a bottle or container, the recycling operator finds it easier to sort plastics by material type.

12/3,K/15 (Item 1 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB

10561865 SUPPLIER NUMBER: 21217841 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Shared data: a ticking Y2K time bomb. (avoiding flawed-data resulting from the year 2000 computer problem)(Column)
McCarthy, Shawn P.
Logistics Management Distribution Report, v37, n9, p99(1)
Sept, 1998
DOCUMENT TYPE: Column LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 660 LINE COUNT: 00053

TEXT:
If you've been hustling to **solve** Year 2000 (Y2K) problems **on** your company's computers, you may feel you've met your goals if you have a schedule in **place** for analyzing lines of **code** so you can identify and fix bad date references.

12/3,K/16 (Item 2 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2003 The Gale Group. All rts. reserv.

10343230 SUPPLIER NUMBER: 20949533 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Visio Maps leaves users lost. (new product from Visio Corp.)(Evaluation)
Cunningham, Cliff
Computer Dealer News, v14, n24, p40(1)
June 22, 1998
DOCUMENT TYPE: Evaluation ISSN: 1184-2369 LANGUAGE: English
RECORD TYPE: Fulltext
WORD COUNT: 747 LINE COUNT: 00059

... a United States map easily (yes, Canadian maps are included). Graduated color or symbol maps are alternate options.
Two problems remained, however. Of the 512 **places** I was importing by zip **code** , 51 were "not found." No further explanation was offered, no list of unknown zip codes was printed, and no method of checking these was available. The other problem was even more trying. **While** the dots were placed **correctly** , clicking **on** a dot elicited no information from the database. Knowing who or what a data point represents on the map would be most useful.
These problems...

12/3,K/17 (Item 3 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2003 The Gale Group. All rts. reserv.

09329020 SUPPLIER NUMBER: 19075482 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Bar coding tracks herbicide distribution.
Falkman, Mary Ann
Packaging Digest, v33, n16, p56(2)
Dec, 1996
ISSN: 0030-9117 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 1763 LINE COUNT: 00139

... the lot number. This insures that, should the case have to be pulled off the pallet load for any reason, a replacement case can be **added** by scanning the small bar **code** , validating it against the pallet manifest, and verifying **that** it has landed **on** the **correct** pallet.
From there, the cases convey to palletizing, passing a Banner accoustic proximity sensor that alerts the second Prestek labeler to produce the pallet license...

12/3,K/18 (Item 4 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2003 The Gale Group. All rts. reserv.

08137244 SUPPLIER NUMBER: 17334243 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Unify dispersed development teams. (synchronization of software development teams) (includes related article on McDonnell Douglas use of object oriented software design)
Baum, David
Datamation, v41, n15, p37(3)
August 15, 1995
ISSN: 1062-8363 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 2148 LINE COUNT: 00184

...ABSTRACT: teams utilizing metasoftware increase productivity by incorporating an environment that maintains all of the software development team's information. Team managers often experience difficulty ensuring **that** programmers are working **on** the **correct** piece of code. Throughout many stages of development, managers strive to implement an effective management structure. Application development tools usually provide sufficient team development capabilities...

...a particular file or object from the same source set. The technology also provides for automated version control and configuration management. Users may want to **add** multiple SCM servers in enhance **code** accessibility.

12/3,K/19 (Item 5 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2003 The Gale Group. All rts. reserv.

07291523 SUPPLIER NUMBER: 15405554 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Specificity of subsidy benefits in U.S. Department of Commerce countervailing duty determinations.
Ragosta, John A.; Shanker, Howard M.
Law and Policy in International Business, 25, n2, 639-683
Wntr, 1994
ISSN: 0023-9208 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 21287 LINE COUNT: 01719

... the factors, a program is specific. As the law continues to develop in this area, practitioners face increasingly complex issues. The courts, Commerce, or Congress **should** provide additional guidance **on** how such issues will be **resolved** . This Article provides a framework for such guidance. * Mr. Ragosta,J.d. 1984, University of Virginia, a partner in the Washington, D.C. office of...

...VI, XVI and XXIII, done Apr. 12, 1979, art. 11, para. 3, 31 U.S.T. 513, 533, T.I.A.S. No. 9619 (emphasis **added**) [hereinafter GATT Subsidies **Code**]. Thus, specificity is not currently a requirement of the GATT. Cf. Judith Hippler Bello & Alan F. Holmer, Subsidies and Natural Resources: Congress Rejects a Lateral...

12/3,K/20 (Item 6 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2003 The Gale Group. All rts. reserv.

06411352 SUPPLIER NUMBER: 13525921 (USE FORMAT 7 OR 9 FOR FULL TEXT)
UPS, truly a business of 'sorts'; an accurate and speedy sorting and delivery system is vital for survival in the next-day express business. (United Parcel Service)
Ditter, Al
Air Cargo World, v83, n2, p38(3)
Feb, 1993
ISSN: 0745-5100 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 2144 LINE COUNT: 00161

... odd-sized and - shaped shipments) are hand carried from the belts to their destination containers, while regulars and smalls go through a succession of sorts **that** ultimately takes them to the **correct**

containers.

Once **on** their way, the regulars and smalls are fed through a primary sort that splits them eight ways by state and the first three digits of the ZIP code. A secondary sort splits them five ways by state and the full five digits of the ZIP **code** , with the smalls being **placed** in boxes and onto a belt that takes them to designated containers.

"The sorters know the state, city and ZIP code

12/3,K/21 (Item 1 from file: 624)
DIALOG(R)File 624:McGraw-Hill Publications

0048978
PA. HIGH COURT REVERSES PUC RULINGS THAT OKAYED CANCELED PLANT RECOVERY
Electric Utility Week October 26, 1987; Pg 11
Journal Code: EUW ISSN: 0046-1695
Word Count: 379 *Full text available in Formats 5, 7 and 9*

TEXT:
... of its rate base or by converting them into operating expenses through amortization" (Dockets 33 and 34, W.D., 1986).

Pankiw noted that the legislature **added** a section to the **code that** does allow the **recovery** of costs but no return **on** the investment if the decision to cancel was prudently made. In this case, however, the cancellations happened before the October 1985 effective date of Section...

12/3,K/22 (Item 1 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)

00927964 95-77356
Microsoft debuts low-cost NT 3.5
Darrow, Barbara; Sperling, Ed
Computer Reseller News n596 PP: 1, 253 Sep 19, 1994
ISSN: 0893-8377 JRNL CODE: CRN
WORD COUNT: 475

...TEXT: and full interoperability, it's hard to move this through the channel."

The training and seeding efforts coincide with the latest release of Windows NT, **code** -named Daytona. The new version **adds** full support for Object Linking and Embedding 2.0, a new TCP/IP stack that allows users to log onto the Internet, a dynamic host configuration protocol and an automatic **recovery** facility **that enables** users to reboot and debug an application while the server is running.

Richard Tong, general manager of Microsoft's Business Systems Division, said the company...

12/3,K/23 (Item 2 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)

00588914 92-04087
Comments on Proposed Earnings Stripping Regulations Under Section 163(j)
Anonymous
Tax Executive v43n6 PP: 425-430 Nov/Dec 1991
ISSN: 0040-0025 JRNL CODE: TXE

...ABSTRACT: comments with the IRS concerning proposed regulations under section 163(j) of the Internal Revenue Code, which relates to earnings stripping. Section 163(j) was **added** to the **Code** to prevent the possible erosion of the US tax base by the use of excessive deductions for

interest paid by a taxable corporation to a...

... in, first out) recapture amount should be deleted because such changes do not reflect cash flow. 4. The anti-rollover rule is ill-advised and **should** be eliminated. 5. The **fixed** stock write- **off** method should be expanded to include other stock purchases. ...

12/3,K/24 (Item 3 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)

00372300 87-31134
On Inventory Capitalization Regulations
Anonymous
Retail Control v55n6 PP: 21-27 Aug 1987
ISSN: 0034-6047 JRNL CODE: REC

ABSTRACT: Section 263A of the Internal Revenue **Code , added** to the **Code** by Section 803 of the Tax Reform Act of 1986, provides comprehensive capitalization rules for certain costs allocable to real or tangible personal property produced...

... in order to focus attention on those areas that should be revised. For example, while Treasury Regulation Section 1.263A-1T(2)(v)(F) provides **that** depreciation and cost **recovery** allowances **on** equipment and facilities placed in service but temporarily idle need not be capitalized, NRMA believes this exclusion should be expanded to include other costs applicable...

12/3,K/25 (Item 1 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext

01026807 CMP ACCESSION NUMBER: CRN19940919S0001
PUSH: Free training, seed units for resellers - MICROSOFT DEBUTS LOW-COST NT 3.5
BARBARA DARROW & ED SPERLING
COMPUTER RESELLER NEWS, 1994, n 596, PG1
PUBLICATION DATE: 940919
JOURNAL CODE: CRN LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: NEWS
WORD COUNT: 477

... and full interoperability, it's hard to move this through the channel.''

The training and seeding efforts coincide with the latest release of Windows NT, **code** -named Daytona. The new version **adds** full support for Object Linking and Embedding 2.0, a new TCP/IP stack that allows users to log onto the Internet, a dynamic host configuration protocol and an automatic **recovery** facility **that enables** users to reboot and debug an application while the server is running.

Richard Tong, general manager of Microsoft's Business Systems Division, said the company...

12/3,K/26 (Item 1 from file: 674)
DIALOG(R)File 674:Computer News Fulltext

079225
Web monitoring tools help IT rest easier
Byline: SUZANNE GASPAR
Journal: Network World
Publication Date: November 08, 1999
Word Count: 855 Line Count: 78

Text:
... or a better way to monitor and manage the Web site.Sanders solved the problem by buying Platform Computing's SiteAssure, a Web monitoring tool **that** takes **corrective** action based **on** predefined policies. He uses it in conjunction with Freshwater Software's SiteScope real-time Web-server monitoring tool to keep the Web site up and...

... next stepWhen Sanders initially assessed the monitoring and management needs of his Web data center, he recognized its immaturity. He planned to first stabilize his **code** base and platform, then **add** products to take corrective action. Sanders sees Web management as a layering process: first encounter the problem and examine frequency, then look at the trend...

12/3,K/27 (Item 2 from file: 674)
DIALOG(R)File 674:Computer News Fulltext

078728
Experts warn of security hole in Microsoft Java machine
Byline: Sharon Machlis
Journal: Network World
Publication Date: October 15, 1999
Word Count: 266 Line Count: 25

Text:
... and the Eudora e-mail program. Karsten Sohr at the University of Marburg reported finding the bug in JVM'sbytecode verifier. The glitch allows a **code** sequence to be **put** together that improperly puts the values from one Java type into the values of another Java type. Bytecode is the name for compiled Java programs...

... is not aware of any users being affected by the problem, the spokesman added. Still, the company takes such security matters seriously, she said. Information **on** a **fix** **should** be available **on** Microsoft's Java Web site.

12/3,K/28 (Item 1 from file: 696)
DIALOG(R)File 696:DIALOG Telecom. Newsletters

00593760
THE MILLINIUM BUG: IT'S NOT TOO LATE FOR EDI USERS Repair Costs Could Reac h More Than $70 Billion
EDI NEWS
March 2, 1998 VOL: 12 ISSUE: 5 DOCUMENT TYPE: NEWSLETTER
PUBLISHER: PHILLIPS BUSINESS INFORMATION
LANGUAGE: ENGLISH WORD COUNT: 1281 RECORD TYPE: FULLTEXT

TEXT:
...problem, Reilly says, you might be able
to archive your legacy data and write a program to handle the dates.
A quick solution is to **add** **code** that reads the year designation
from six-digit legacy data. If the year is 50 or below, it is assumed
to be in the 21st **on** active applications **that** cannot be
fixed and must be replaced prior to 1999; choose active applications
whose repairs or replacement can be delayed beyond 2000. and finally,
dormant applications whose repairs...
?

15/3,K/1 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

02267455 SUPPLIER NUMBER: 53728459 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Q&A C++. (question and answer). (Column)
DILASCIA, PAUL
Microsoft Systems Journal, 14, 3, 59(1)
March, 1999
DOCUMENT TYPE: Column ISSN: 0889-9932 LANGUAGE: English
RECORD TYPE: Fulltext
WORD COUNT: 3222 LINE COUNT: 00307

... asked me this question, and I recently bumped into the same problem as well. Normally when there's a works-in-debug, fails-in-release **problem** , the first thing to look for is some **code** you accidentally **put** in an **ASSERT** statement, such as:
ASSERT((x=DoSomething()) !=5);
Here, the assignment x=DoSomething is omitted in release builds since ASSERT evaluates to a no-op. The...

15/3,K/2 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

02236559 SUPPLIER NUMBER: 53183530 (USE FORMAT 7 OR 9 FOR FULL TEXT)
MICROSOFT TURNS THE SCREW ON TEVANIAN TESTIMONY.
Computergram International, 3532, NA
Nov 5, 1998
ISSN: 0268-716X LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 658 LINE COUNT: 00053

TEXT:
...of its author. Tevanian was taken to task over his allegations that Redmond tried to "sabotage" Apple's QuickTime technology. Edelman also questioned Apple's **assertions** that Microsoft created "misleading error messages" to deprive QuickTime of the opportunity of processing certain multimedia file types, and that Apple had only limited success...

...other problems had been fixed. He added that Apple "took the time in order to get good data" to respond to Microsoft. He also asserted **that** he felt the onus was **on** Microsoft to **fix problems** with their **software** . Edelman asked if Tevanian really thought that the bugs had been "cooked up" by Re dmond. Tevanian responded curtly with this question, "if they could
...
...sure if they were intentional or not." Edelman then stated that as Tevanian had not attended internal Microsoft meetings, he could have "no basis to **assert** " that the error messages were intentionally put there to harm QuickTime. Tevanian asked, "what other goal could there have been?" But went on to say...

15/3,K/3 (Item 3 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

02022325 SUPPLIER NUMBER: 18962638 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Visual FoxPro 5.0. (Microsoft's DBMS) (Software Review) (Evaluation)
Granor, Tamar E.
Data Based Advisor, v15, n1, p30(5)
Jan, 1997
DOCUMENT TYPE: Evaluation ISSN: 0740-5200 LANGUAGE: English
RECORD TYPE: Fulltext; Abstract
WORD COUNT: 4138 LINE COUNT: 00319

... extremely valuable tool for discovering which parts of an application are slowing things down.

Finally, assertions have been added to the language. The new ASSERT **command** lets you **put** assumptions in the **code** to be checked at runtime. If an **assertion fails** , a message from the **ASSERT** command is displayed. **Assertion** checking can be toggled on and off.

Team development

There are several ways in which VFP 5 is a better tool for team development than...

15/3,K/4 (Item 4 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01663177 SUPPLIER NUMBER: 14618566 (USE FORMAT 7 OR 9 FOR FULL TEXT)
C/C++ Questions & Answers. (Column)
DiLascia, Paul
Microsoft Systems Journal, v9, n1, p83(4)
Jan, 1994
DOCUMENT TYPE: Column ISSN: 0889-9932 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 2428 LINE COUNT: 00198

... contains the following lines

HWND hWndCtrl = (HWND)LOWORD(lParam); : : ASSERT (::IsWindow(hWndCtrl));

If your library uses nonstandard arguments to WM[underscore]COMMAND, the assertion will **fail** . If the **code** is not well-behaved, you'll have a hard time running it hooked up to MFC. You'll have to find each glitch by trial and error, and overload the appropriate virtual functions. In the preceding example, you'd have to rewrite OnCommand without the **ASSERT** checks. Of course, the best way is to write the **code** correctly in the first **place** (this is particularly important for portability) but obviously you can't do that for code that you didn't write such as a commercial library...

15/3,K/5 (Item 5 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01551657 SUPPLIER NUMBER: 12236786 (USE FORMAT 7 OR 9 FOR FULL TEXT)
User report: debugging with Sherlock. (Sherlock debugging software for C programming language) (Tutorial)
Ream, Edward K.
C Users Journal, v10, n6, p121(3)
June, 1992
DOCUMENT TYPE: Tutorial ISSN: 0898-9788 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 1481 LINE COUNT: 00118

I wrote a set of programming tools, called Sherlock, which tackles the **problems** caused by **adding** debugging **code** to **programs** . With Sherlock, debugging **code** can be enabled or disabled without recompiling and relinking your program. Inserted printf statements trace your functions and data structures exactly as you desire. You can **insert** heavy-weight **assertions** -- detailed **code** to check your data structures -- wherever you wish with no unwanted time penalty. Finally, Sherlock measures program performance at whatever level you desire.

Sherlock contains...

15/3,K/6 (Item 6 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01540718 SUPPLIER NUMBER: 12684752 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Debugging with assertions. (Tutorial)
Bates, Rodney M.

C Users Journal, v10, n10, p40(7)
Oct, 1992
DOCUMENT TYPE: Tutorial ISSN: 0898-9788 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 4243 LINE COUNT: 00319

... The assert macro takes a single argument in parentheses. This argument is a truth-valued expression that you believe will always be true at this **place** in your **code** . It expands into **code** that checks the expression and, if it turns out **FALSE** , prints a message on standard output and terminates the **program** . The message usually looks something like

Assertion **failure** I <= 0 file: mergetxt.c line: 1073

This means that the assertion I <= 0 on line 1073 of source file mergetxt.c has failed. Some...

15/3,K/7 (Item 7 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01525975 SUPPLIER NUMBER: 12344522 (USE FORMAT 7 OR 9 FOR FULL TEXT)
A Windows assert() with symbolic stack trace. (includes related article on sanity checks for .sym files) (Tutorial)
Pietrek, Matt
Windows-DOS Developer's Journal, v3, n7, p49(10)
July, 1992
DOCUMENT TYPE: Tutorial ISSN: 1059-2407 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3847 LINE COUNT: 00299

ABSTRACT: Assertions are vital to writing robust and correct code. The assert() macro in C and C++ allows the programmer to **place** assertions in the **code** and switch **assertion** checking on or off before compiling. One **problem** with **assert** () is that the low-level information provided when the **assertion** fails is often of little direct use. Instruction is given for building a version of **assert** () for Microsoft Windows programs; this version supplies a stack trace with sufficient symbolic information so that the programmer can see what was transpiring when the **assertion** failed. The key to this version of **assert** () is the .sym file, which makes symbolic stack tracing possible. Many debugging tools use .sym files. A programmer performing a symbolic stack trace needs a...

15/3,K/8 (Item 8 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2003 The Gale Group. All rts. reserv.

01446929 SUPPLIER NUMBER: 11047844 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Reuse implies Eiffel. (Applied Logic Research's Eiffel, an object-oriented programming language) (technical)
Steggles, Pete
EXE, v6, n1, p39(5)
June, 1991
DOCUMENT TYPE: technical ISSN: 0268-6872 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2984 LINE COUNT: 00220

... editing, with errors almost certain. You think it can't get worse? It can! Think how we usually learn how to use a piece of **software** : when we're learning it we make **errors** and we rely on good **error** -reporting in the **software** to tell us what we've done wrong. in fact, if the error-reporting is bad, or not there, then we give up using the...

...a client, then we're going to need good error-reporting. So in future, if we want to have significant reuse, programmers will have to **put** good-quality **error** reporting **code** into all their class definitions. So, to implement reuse in a normal object-oriented language, each class needs

high quality documentation and a sophisticated error...
...to write the code. ironically, trying to implement reuse with an inadequate language could easily reduce productivity; it could be more effective to throw most **code** away. Eiffel to the rescue These were the **problems** Meyer was faced with. He was fortunate in having an understanding of formal methods of software development, and in seeing how to apply a simple...

...like VDM and Z, to solve all the problems at once. Meyer's solution is to recognise that classes define state machines. He provides an **assertion** mechanism which lets us say exactly when a function call can be performed and, if it can, what its execution does to the state. it does this by allowing us to define, for each function, preconditions and postcondition& These are two sets of **assertions** about the state of the object whose function is being called. The preconditions are the **assertions** which must be true for the function to mean anything, the postconditions the **assertions** which the function guarantees to be true when it has terminated. Additionally, Eiffel provides a way of defining 'class invariants', **assertions** which are always true of the class which possesses them. What does this buy us? On its own, not a lot. But, combined with a few tools, everything we need. Two tools provided with the Eiffel environment allow us to extract the appropriate **assertions** to provide automatically- generated documentation for a class in terms of pre- and post-conditions and invariants. The program f 1 at produces a flat...

...been programmed without inheritance. short takes a class definition and strips out all the imperative code, all the non-exported function definitions and all the **assertions** involving nonexported features to give a description of the abstract data type which the class characterises. If **assertions** are used well, the combination of flat and short can be used to produce high-quality documentation for a class. in particular, with flat we ...

...of inheriting documentation while automatically accounting for renaming and redefinition. What's more, the documentation is checkable. The compiler provides options for checking that the **assertions** are true when they ought to be, ie when a function is called its precondition is checked (and so is the class invariant) and when it exits, the postcondition is checked (and the class invariant again). If an **assertion** is violated, then the Eiffel system generates a detailed and informative error message describing which **assertion** was violated and printing the feature call stack as it was when the **assertion** was violated. Figure 1 shows part of the code for a ring menu (a sort of menu which 4GIs tend to provide). From Figure 2...

...banner. Should we go wrong, die system will tell us which **assertion** we violated. It takes a while to realise just how much we gain from **assertion** checking, and the flat and short **programs** . By including the appropriate **assertions** we get a test suite, **error** handler and inheritable, checkable documentation for free. The icing on the Cake There is not enough space to describe the other benefits of the Eiffel...

...type system, genericity, expanded types, generalised iterator classes and switchable garbage collection. Suffice it to say that Eiffel isn't just a solution to one **problem** , but is a serious **software** engineering language. The Eiffel environment provides more than just flat and short, of course. Included in the package are: an incremental compiler which calculates file...

15/3,K/9 (Item 9 from file: 275)

01295846 SUPPLIER NUMBER: 07242466 (USE FORMAT 7 OR 9 FOR FULL TEXT)
'Dear professor.' (writing low-maintenance software)
Glass, Bob
Software Magazine, v9, n5, p8(2)
April, 1989

ISSN: 0897-8085 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 809 LINE COUNT: 00066

...ABSTRACT: should be done in one place. Examples are data abstraction, modular programming, file-driven design, and object-oriented programming. Defensive design includes capabilities in the **software that enable** it to **recover** from any **problem** . Examples include **assertions** , exception handling, fault tolerance, and capacity planning. Improved maintenance tools and appropriate management are also deemed important.

15/3,K/10 (Item 10 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01294661 SUPPLIER NUMBER: 07224290 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Program laborious. (discussion on exploitation of parallelism inherent in a given program)
Emrath, Perry
UNIX Review, v7, n4, p51(7)
April, 1989
ISSN: 0742-3136 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3897 LINE COUNT: 00317

... those assertions each time the program is ported to another system.
The amount of time parallelizing compilers actually require to analyze, parallelize, and compile a **program** is also a serious **problem** . Clearly, such parallelizing compilers use many more CPU cycles than do "sequential" compilers (that is, compilers that generate sequential code). Furthermore, programmers who use a should have been parallelized but wasn't. They may then **add assertions** or change the **code** slightly and do another compiler run to see if the desired effect has been achieved. In practice, getting an existing code compiled into efficient parallel...

15/3,K/11 (Item 1 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

03102380 Supplier Number: 46347719 (USE FORMAT 7 FOR FULLTEXT)
AUTODESK TRIES AGAIN
Computer Aided Design Report, v5, n6, pN/A
May 1, 1996
Language: English Record Type: Fulltext
Document Type: Newsletter; Trade
Word Count: 4009

... that the problems with AutoCAD version 13 were corrected with the fifth and current version which Autodesk calls the C4 patch.
Unfortunately, the company's **assertions** aren't born out by experience. During our sessions with the Mechanical Desktop, AutoCAD crashed several times for no apparent reason. Most of these fatal errors occurred when doing routine housekeeping functions, such as changing layer colors, rather than when creating or modifying solid geometry. A Detroit-area engineering company **that** uses the C4 **patch on** more than 60 computers also told us it has experienced many unexplained fatal **errors** when using the **software** . According to a supervisor, the **problem** appears to occur when a series of commands is entered from a tablet too quickly.
Besides problems with core AutoCAD, we also observed strange behavior ...

15/3,K/12 (Item 1 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

07673405 Supplier Number: 63649396 (USE FORMAT 7 FOR FULLTEXT)

Quotable highlights from the antitrust trial.(Company Business and Marketing)
Haney, Clare
Network World, pNA
June 12, 2000
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; General Trade
Word Count: 4270

... monopoly." - from the Jusitice Department's rebuttal to Microsoft's proposed conclusions of law. (Jan. 26) </p> </p> "Microsoft quotes several findings to support the **assertion** that there are certain benefits to the commingling of Internet Explorer and Windows. But this court's findings plainly say the opposite: There are no...the disruption to its business that will inevitably result from having the threat of a breakup hang over its head like the sword of Damocles **while** this Court conducts proceedings **on remedies** . Not only might Microsoft lose irreplaceable employees, but third parties may be unwilling to enter into routine business agreements with Microsoft while its continued corporate...
...vice president for law and corporate affairs on the government's proposal that Microsoft should make public key elements of its Windows operating system source **code** . (May 10) </p> "The **problem** that Microsoft is facing is that the court just doesn't trust them given their behavior under the last consent decree. Against the court's...

15/3,K/13 (Item 2 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

03952613 Supplier Number: 45724817
AT&T to oppose 972 area code plan
Dallas Morning News (TX), pD1
August 11, 1995
Language: English Record Type: Abstract
Document Type: Newspaper; Trade

ABSTRACT:
...to be part of a Southwestern Bell hearing to be held in 09/95. An additional code in the region would require dialing an area **code** prior to the seven-digit phone number, among other **problems** , but Southwestern Bell exerts that lack of available numbers for the 214 code prompts the move. The opposing parties' **assertion** is to **add** a new area **code** based on a geographic split. ...

15/3,K/14 (Item 1 from file: 624)
DIALOG(R)File 624:McGraw-Hill Publications

01109504
A COMPLAINT THAT NYISO WRONGFULLY DENIED NIAGARA MOHAWK ENERGY MARKETING INC.'S ATTEMPT TO EXPORT POWER
Inside FERC August 7, 2000; Pg 17; Vol. 21, No. 32
Journal Code: FERC ISSN: 0-163-948X
Section Heading: ELECTRIC TRANSMISSION
Word Count: 351 *Full text available in Formats 5, 7 and 9*

TEXT:
... manual check procedures may be impractical at this point, the July 26 order went on to state that ``since the NYISO is currently implementing its **software** fix of this **problem** , it would be an inefficient use of the commission's and the affected parties' limited resources to now develop a compensation mechanism.''
Instead, the order...

15/3,K/15 (Item 1 from file: 15)

DIALOG(R)File 15:ABI/Inform(R)

02164940 72672578
State accountability for violations of intellectual property rights: How to "fix" Florida prepaid (and how not to)
Berman, Mitchell N; Reese, R Anthony; Young, Ernest A
Texas Law Review v79n5 PP: 1037-1197 Apr 2001
ISSN: 0040-4411 JRNL CODE: TRX
WORD COUNT: 83463

...TEXT: Takings, and the Florida Prepaid Decisions

College Savings Bank and Florida Prepaid both involved the State of Florida's operation of a college tuition prepayment **program** under the catchy name of the "Florida Prepaid Postsecondary Education Expense Board." College Savings Bank had previously patented a similar business methodology. Alleging that the... of its Eleventh Amendment immunity and subject to suit in federal court under the federal intellectual property laws. Because Congress has some latitude to provide **remedies that** go beyond what the Constitution might require directly, so long as those remedies are directed at an actual constitutional violation,240 we think the offending... owner to sue the state official responsible for infringement in federal court in her official capacity for prospective injunctive relief, regardless of the state's **assertions** of sovereign immunity.285 Although the claim in Ex parte Young involved activity by the state that was alleged to be in violation of the...question whether an action in federal court by a qui tam relator against a State would run afoul of the Eleventh Amendment," the majority's **assertion** of "'a serious doubt' on that score"408 does not bode well for broader proposals like Professor Siegel's, which raise more difficult problems than...public ones443-seems likely to hold true in most intellectual property situations.

We thus conclude that, under current law, state officials are probably entitled to **assert** a qualified immunity defense when sued under the federal intellectual property laws.444 Congress could, however, alter that situation by statute. We discuss whether Congress...Progress of Science and useful Arts."480 The funding does so by directly subsidizing research; the condition does so by removing the dampening effect that **assertions** of sovereign immunity have on other would-be innovators.481 By way of contrast, imagine that Congress demanded waiver of sovereign immunity in intellectual property...

15/3,K/16 (Item 2 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)

02019394 53818603
Efficient run-time assurance in distributed systems through selection of executable assertions
Schollmeyer, Martina; McMillin, Bruce
Journal of Systems & Software v52n1 PP: 33-54 May 15, 2000
ISSN: 0164-1212 JRNL CODE: JSS

...ABSTRACT: can be obtained by comparing, at run-time, the actual behavior of a program with the expected behavior described in the program's specification. Executable **assertions** , **embedded** into the **program code** , can determine when there are discrepancies between actual and expected behavior. Temporal subsumption is introduced to remove, from a given set of **assertions** for a specific distributed program, the **assertions** which perform redundant checking. The remaining set of **assertions** is then the set necessary to provide run-time assurance. To subsume **assertions** , the flow graphs of the individual components of the distributed system are examined using a graph traversal algorithm. Temporal subsumption is a pre-processing step that creates a smaller set of **assertions** to be embedded into the **program** and to be checked at run-time. This makes **error** detection at run time less time-consuming and thus more efficient.

15/3,K/17 (Item 3 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)

00222034 84-00595
Living with Hard Realities
Nathan, Richard P.
Planning v49n9 PP: 5-10 Oct 1983
ISSN: 0001-2610 JRNL CODE: PLN

ABSTRACT: Current **assertions** that government programs do not work and **that** reliance **should** be placed **on** the private sector to solve public **problems** are based **on** the belief that government **programs** can harness private energies to stimulate the economy, create jobs, aid distressed areas and industries, and enhance the tax base. ''Economic development'' is here discussed...
?